FLEET TECHNOLOGY LIMITED
A FLEET AEROSPACE COMPANY

FLEET TECHNOLOGY LIMITED 1982C

DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLLED TWO-DIMENSIONAL ACOUSTIC ARRAY FOR IMAGE DEFECTS IN OFFSHORE STRUCTURE WELDMENTS

PHASES 1 AND 2

WORK ON THIS PROJECT WAS FUNDED BY THE METALS TECHNOLOGY LABORATORIES, CANADA CENTRE FOR MINERAL AND ENERGY TECHNOLOGY, ENERGY MINES AND RESOURCES CANADA.

DSS FILE NUMBER: 03SQ.23440-5-9165

DSS CONTRACT SERIAL NUMBER: OSQ85-00304

CANMET SCIENTIFIC AUTHORITY: DR. D.K. MAK

SUBMITTED TO:

ENERGY, MINES AND RESOURCES
CANMET, METALS TECHNOLOGY LABORATORIES
568 BOOTH STREET
OTTAWA, ONTARIO
KIA OG1

SUBMITTED BY:

FLEET TECHNOLOGY LIMITED
311 LEGGET DRIVE
KANATA, ONTARIO
K2K 1Z8

APRIL, 1990

177pp

MICROMEDIA

91-00720/

ARCTEC CANADA LIMITED 1982C

DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLLED TWO-DIMENSIONAL ACOUSTIC ARRAY FOR IMAGE DEFECTS IN OFFSHORE STRUCTURE WELDMENTS

PHASE 1

INTERIM REPORT BY:

R. BROWN, ACL.
H. JONES & ASSOCIATES

WORK ON THIS PROJECT WAS FUNDED BY THE PHYSICAL METALLURGY RESEARCH LABORATORIES, CANADA CENTRE FOR MINERAL AND ENERGY TECHNOLOGY, ENERGY MINES AND RESOURCES CANADA.

DSS FILE NUMBER: 03SQ.23440-5-9165

DSS CONTRACT SERIAL NUMBER: OSQ85-00304

CANMET SCIENTIFIC AUTHORITY: DR. D.K. MAK

SUBMITTED TO:

ENERGY, MINES AND RESOURCES
CANMET, PHYSICAL METALLURGICAL RESEARCH LABS
555 BOOTH STREET
OTTAWA, ONTARIO
KIA OG1

SUBMITTED BY:

ARCTEC CANADA LIMITED
311 LEGGET DRIVE
KANATA, ONTARIO
K2K 1Z8

MAY, 1987

EXECUTIVE SUMMARY

In this first phase of the project, a 10 x 10 array of acoustic transducers was designed to operate in steel at an operating frequency of 5 MHz. The array was manufactured by cementing 100 wires in a tungsten loaded epoxy backing, gluing a sheet of PZT crystal on top, then cutting the PZT into a matrix of transducers. The electronics for using twenty elements as transmitters and eighty as receivers were designed and built, along with an interface for an IBM PC compatible computer.

Serious problems were encountered in the manufacturing stage of the array, in particular the cutting of the PZT into a matrix resulted in the PZT being pulled off of the epoxy backing. Several different methods were tried but none was very successful. Other than this, the work was successful, the electronics performing adequately.

SOMMAIRE

Dans la première phase de ce projet, une matrice 10 x 10 d'éléments acoustiques a été conçu pour opéres sur l'acuer, à une fréquence de 5 MHz. La matrice à été construite en cémentant 100 fils dans un support de résive époxy chargé de tungstène et collé à une feuille de crystal PZT. La feuille de PZT as ensuite été découpé pour former la matrice. Les circuits élèctroniques pour vingt élèments émitteurs et quatre-vints élèment détecteurs, ainsi qu'une interface pour un micro-ordinateur IBM-PC, ont étè dessiner et construits.

Quelques problèmes sérieux se sont démontrer dans les étapu de manufacture de la matrice. En particular, la fenille de crystal PZT se détacha lors de la découpare de ce dernier. Plusieurs méthodes ont été essayer sans aucun succès absolut. À l'exception de ce problème, tout autre travail a été exécuter avec succès, et les circuits élèctroniques ont performer adéquatement.

TABLE OF CONTENTS

1. INTRODUCTION..1
    - 1.1 Background..1
    - 1.2 Objectives..1
    - 1.3 Overview of present work..1

2. SYSTEM DESIGN..2
    - 2.1 Acoustic transducer design..2
        - 2.1.1 Principle of transducer design..2
        - 2.1.2 Phased array..2
    - 2.2 Electronics design..5
        - 2.2.1 Transmitter subsystem..10
        - 2.2.2 Receiver subsystem..10
        - 2.2.3 IBM I/O interface..10
    - 2.3 Mechanical design..11
    - 2.4 Computer and interfacing..11
    - 2.5 Control and data acquisition software..11

3. SYSTEM CONSTRUCTION AND ASSEMBLY..19
    - 3.1 Construction of transducer head..19
        - 3.1.1 Preparation of the mold..19
        - 3.1.2 Molding of base..20
        - 3.1.3 Attachment of ceramic..20
        - 3.1.4 Cutting ceramic..21
    - 3.2 Construction of electronic components..23
        - 3.2.1 Preamplifier boards..23
            - 3.2.1.1 Preamplifiers..23
            - 3.2.1.2 Multiplexers..24
        - 3.2.2 Transmitter driver cards..24
        - 3.2.3 Oscillator card..24
        - 3.2.4 Control bus card..24
        - 3.2.5 Assembly of electronic card..24
    - 3.3 Mechanical construction..25
    - 3.4 Summary of construction procedures..25
    - 3.5 Problems encountered..25

4. TESTING AND RESULTS..26
    - 4.1 Test procedures..26
    - 4.2 Beam characteristics and assessment of performance......30

5. CONCLUSIONS AND RECOMMENDATIONS..39
    - 5.1 Conclusions..39
    - 5.2 Recommendations..39

APPENDIX A - TRANSDUCER DESIGN
APPENDIX B - DIRECTIVE PATTERN OF TRANSMISSION ARRAY
APPENDIX C - ELECTRONICS CIRCUIT DIAGRAM
APPENDIX D - SOFTWARE LISTINGS
APPENDIX E - A MODEL FOR THE PROPAGATION OF SHORT PULSES OF ULTRASOUND IN A SOLID

1. INTRODUCTION

1.1 Background

Non-destructive testing of materials using ultrasonic energy has been used successfully for some years now, and continues to be an area of much research. Ultrasonic testing equipment, particularly in commercially available systems, has had limitations in the areas of flexibility and ease of use. For instance, systems using single transmit and receive transducers must be repositioned many times to inspect different areas. Fixed transducer characteristics such as the directive pattern, aggravate the amount of work involved.

Relatively simple A scan systems use a CRT type of display that presents reflectivity as a function of depth, the latter being derived from an assumed constant speed of propagation and a linear time sweep. Such displays are much more difficult to interpret than B scan displays that show what appears to be an actual cross sectional image of the material under test. B scan displays are not flexible, however, in the sense that they usually incorporate dedicated hardware to permit real time operation.

1.2 Objectives

The objective of this work is to design and build a two dimensional acoustic array which combined with appropriate software, will permit reconstruction of 3D images of the interior of steel weldments.

By using an array with electronics that can control the relative phasing of transmit elements, the composite beam may be steered. The benefits of this include the reduction of the amount of physical movement of the transducers, and the capability to concentrate the acoustic energy in areas of interest.

Scanning the array's receiver elements and reconstructing the images in software provides a very flexible environment. Lenses can be created mathematically to focus the measured signals, generating a three dimensional grid of values that represent the reflectivity of the object. The 'lens' used can be of any type one can write equations for, and it need not be physically realizable.

Another benefit of software reconstruction is that given an image, there is a wealth of image processing techniques that are available to further enhance the results. These are beyond the scope of this project.

1.3 Overview of Present Work

This report covers the first of the two phases of this project. It is concerned primarily with the design and construction of the acoustic array, the associated electronics, and the control/data acquisition software. Phase II concentrates on the image reconstruction software and testing of the entire system.

2. SYSTEM DESIGN

2.1 Acoustic Transducer Design

2.1.1 Principle of Transducer Design

The scheme of the single transducer is shown in Figure 2.1. The piezoelectric ceramic is covered by conducting layers (silver) which form the electrical input/output port. The left acoustic port is terminated by a back load, or backing medium and the right acoustic port by the medium load via an acoustic impedance transformer formed by quarter-wave acoustic impedance matching layers. The backing medium will not produce an internal reflection so that the energy transmitted into the backing will not be reflected back into the transducer.

The intensity transmission coefficients of an acoustic wave from medium 1 to medium 2 are:

$$T = (4Z_1/Z_2)\,(1 + Z_1/Z_2)^{-2} \qquad (1)$$

where Z_1 and Z_2 are the characteristic impedances (density x speed) of medium 1 and 2. For Z_1/Z_2 tending to 1, T tends to 1, i.e. there is complete transmission when the impedances are matched. If Z_1/Z_2 tends to zero or infinity, T tends to zero, i.e. no transmission.

The characteristic impedances (ρc) of PZT and steel are 29 x 10^6 and 47 x 10^6 kg/m^2s respectively, which means a slight mismatch and hence a little loss of acoustic energy. In order to improve efficiency, a quarter-wavelength matching layer can be used to obtain maximum transmission into the steel. Therefore, a quarter-wave length brass matching layer is used.

The detail of the transducer design is described in Appendix A. A single element of the phased array is 1mm square which is backed by tungsten power loaded epoxy. The operating frequency is 5MHz. Figure 2.2 shows the calculated directivity pattern of a single element. The half-power angle is ±32.5 degrees which is a broad beam pattern.

2.1.2 Phased Array

In this project the transmission and reception of the ultrasound is to be done by a 10 x 10 array of transducers. This array emits sound from twenty of its elements, the relative phasing of which is computer controlled to adjust the beam direction. Eighty receiving elements provide the signals for image reconstruction.

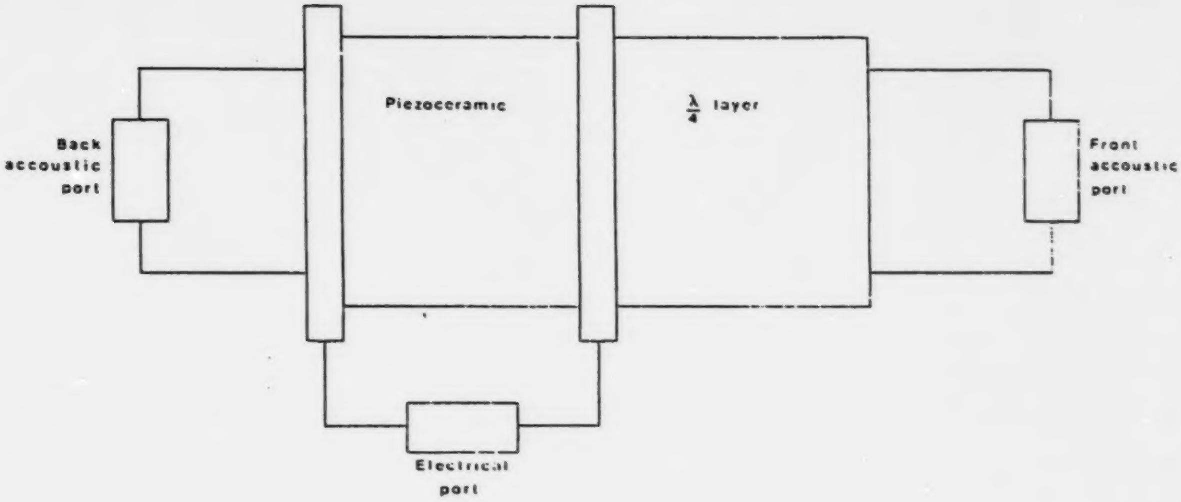
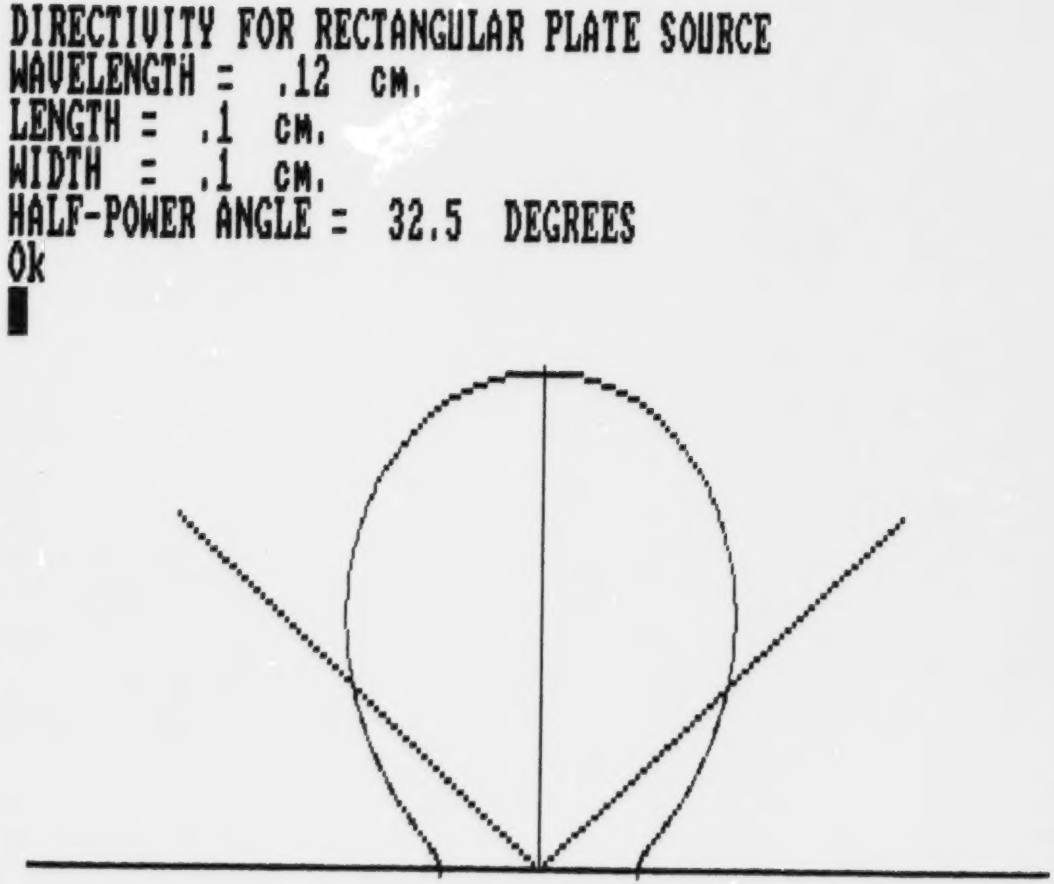

Piezoceramic
$\frac{\lambda}{4}$ layer
Back
accoustic
port
Front
accoustic
port
Electrical
port

Figure 2.1 Simple scheme of a piezoelectric transducer.



```
DIRECTIVITY FOR RECTANGULAR PLATE SOURCE
WAVELENGTH =  .12  CM.
LENGTH =  .1  CM.
WIDTH  =  .1  CM.
HALF-POWER ANGLE =  32.5  DEGREES
Ok
█
```

Figure 2.2 Directive pattern of 1mm x 1mm element.

The primary information obtained from the receivers consists of the arrival time of ultrasound relative to the time of the transmit pulse, the relative amplitudes of the received signals, and the relative phase of each of the received signals. It is very important that these factors be measured accurately and this information be available for subsequent signal processing.

Given the practical restrictions on the number of elements and the overall dimensions, as well as the desire for as narrow a main beam lobe as possible, the configuration of the phased array was selected as shown in Figure 2.3. The spacing between adjacent elements is 1mm. The 20 elements on the two diagonal lines are used as transmitters and the remaining 80 elements are the receivers. Figure 2.4 shows the location of transmitters. The composite beam produced by the transmitters can be steered in different directions by a controlled time delay between adjacent elements. The directive pattern for all elements firing simultaneously is shown in Figure 2.5. The side lobe is approximately at 60° from the main lobe and the amplitude is 30% of the main lobe. The half power angle is ±2.5°, a narrow beam pattern. The detail of the directive pattern can be found in Appendix B.

Figure 2.6 shows a detailed scheme of the phased array assembly. It consists of a piezoelectric ceramic plate with a heavy backing material attached to the rear face and a quarter-wave length brass matching layer attached to the front face. The backing layer consists of a plate of tungsten-loaded epoxy. The tungsten-loaded epoxy is a mix of CIBA resin, hardener and tungsten powder. The proportion of each component is shown in Figure 2.6. The Almax Ai-5 PZT crystal is used. An electrode attached to the PZT is shielded by a hypodermic needle.

2.2 Electronics Design

A nine board electronics system was designed to interface eight digital bits and one analog line to the one hundred element array. The boards were designed to fit in a 6 cm cube with the control signals at one face feeding through seven parallel cards to an analog face. This design consists of a dual bus system, one analog bus for interfacing to the array, and one control bus for providing digital control to the system. Two identical transmitter cards convert digital control codes into twenty high power transmit signals. Five identical receiver cards select one of eighty analog receive lines for analog output. A single DB-25 connector provides all signals to and from the phased array assembly. The details of the circuit design can be found in Appendix C.

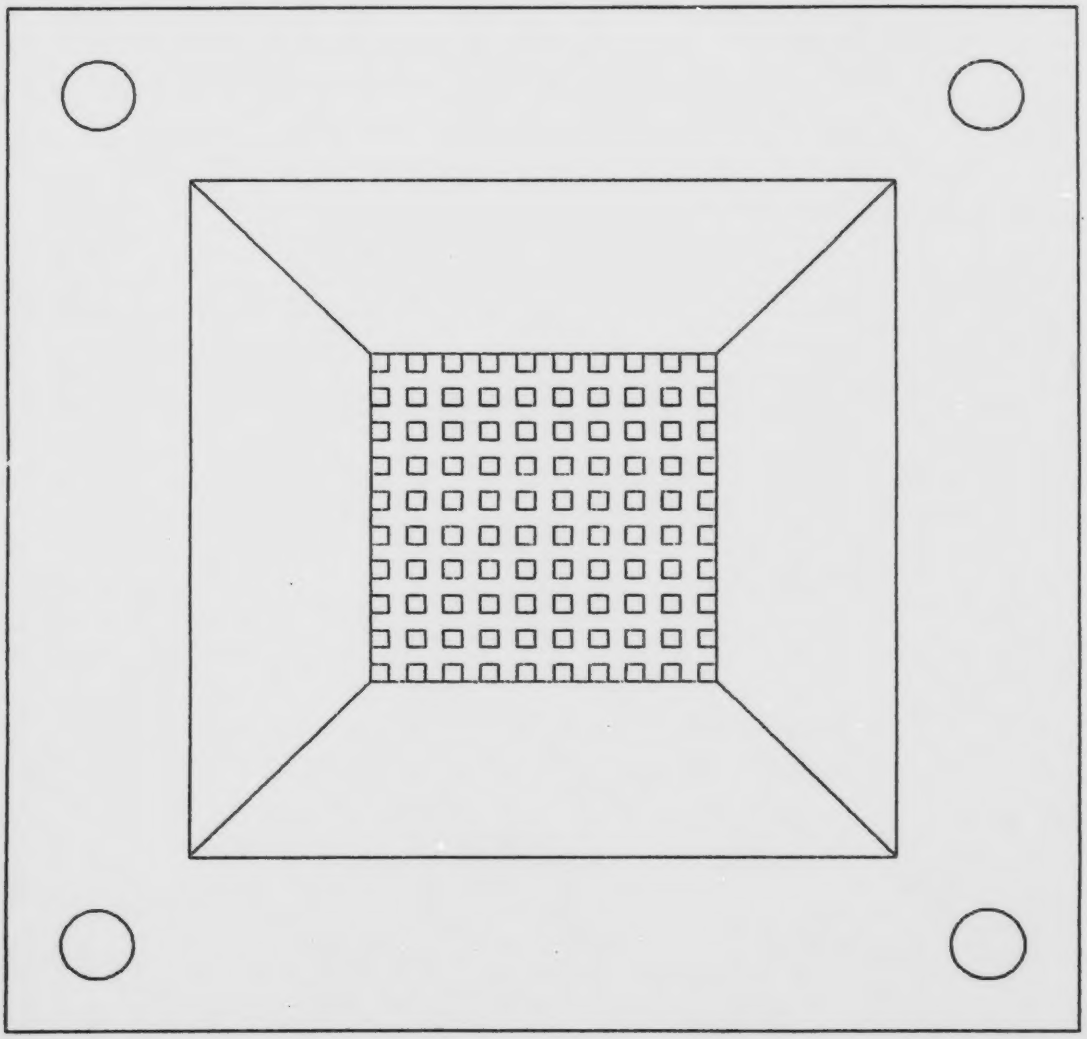


Figure 2.3 Top view of the phased array.

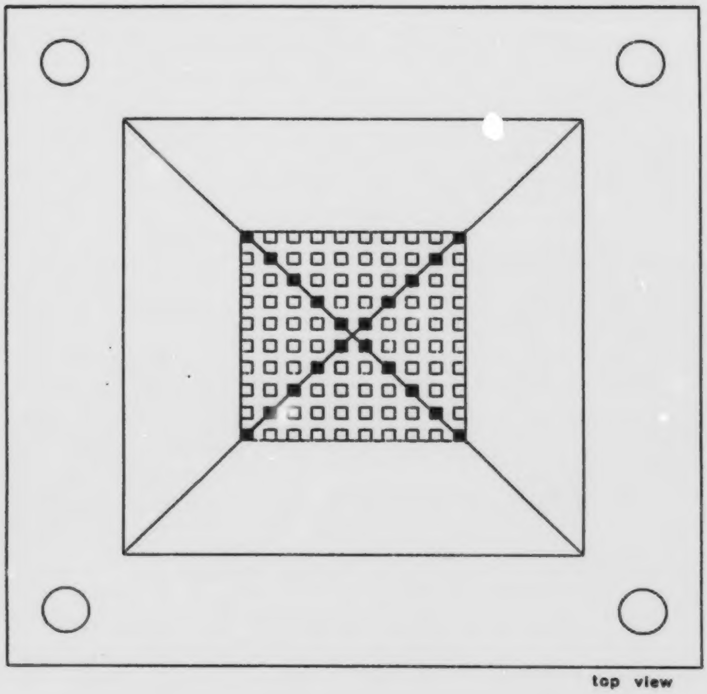
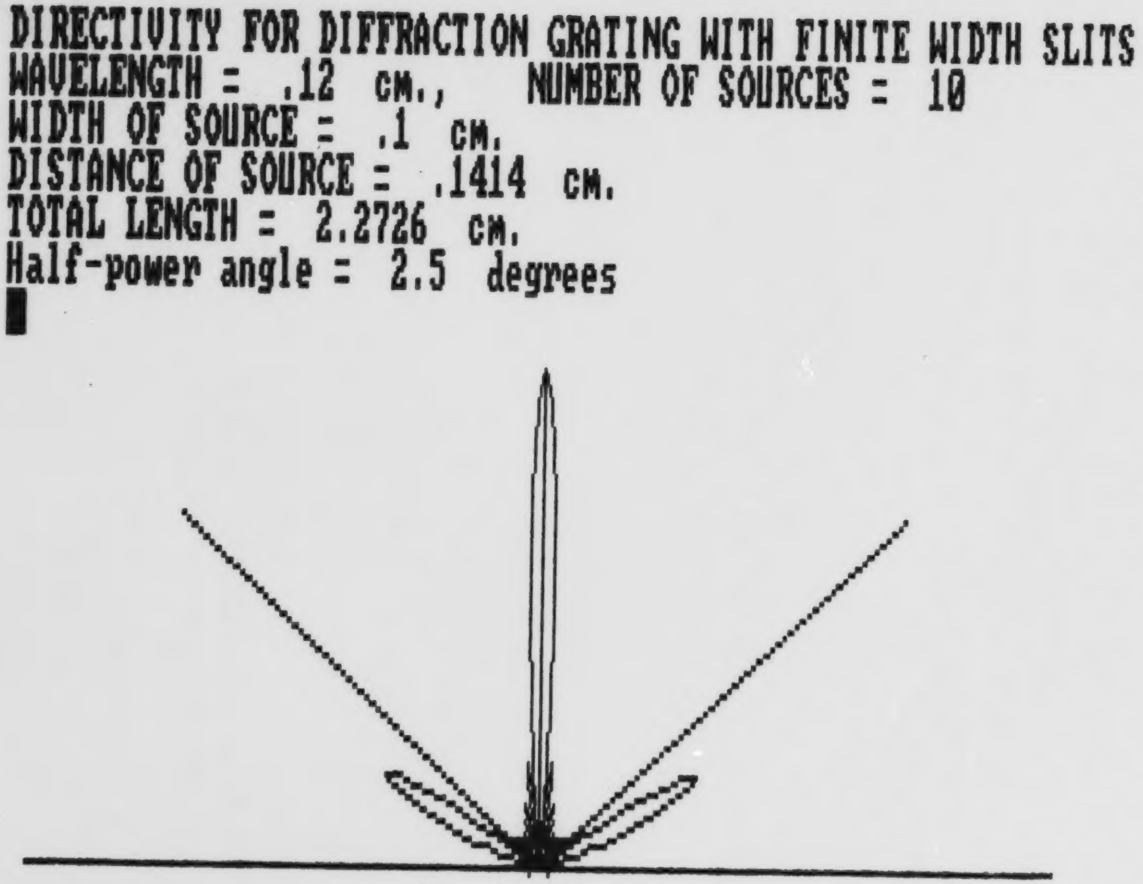
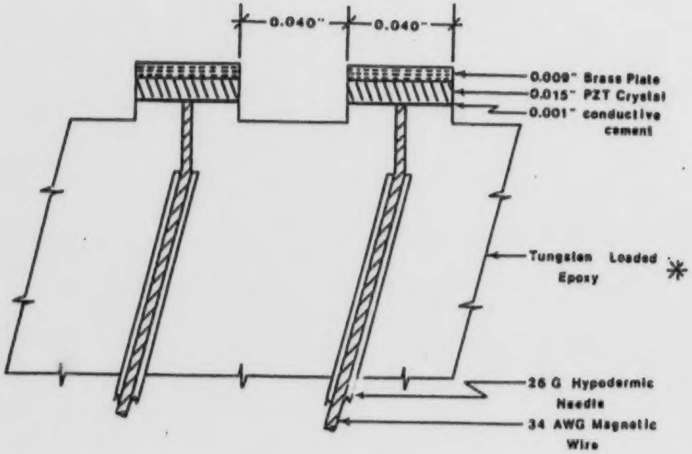


Figure 2.4 Configuration of the transmitting elements on the phased array.



```
DIRECTIVITY FOR DIFFRACTION GRATING WITH FINITE WIDTH SLITS
WAVELENGTH =  .12  CM.,    NUMBER OF SOURCES =  10
WIDTH OF SOURCE =  .1  CM.
DISTANCE OF SOURCE =  .1414  CM.
TOTAL LENGTH =  2.2726  CM.
Half-power angle =  2.5  degrees
```

Figure 2.5 Directive pattern of the transmitting array.


0.040"
0.040"
0.009" Brass Plate
0.015" PZT Crystal
0.001" conductive cement
Tungsten Loaded Epoxy *
26 G Hypodermic Needle
34 AWG Magnetic Wire

* Component	Parts by Weight
Araldite 502	100
Hardener HY956	20
Tungsten Powder	960

Figure 2.6 Detailed structure of the phased array.

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpn	Phased Array Assembly
date Aug 1986	Detail 1 (see page 3)
revised	sheet 4 of 7

2.2.1 Transmitter Subsystem

The heart of the transmitter board consists of a complex high speed TTL switching system utilizing surface mount F (fast) circuitry. Given a direction bit and a four bit phase count, the control portion of the board generates ten TTL pulses, each 100 ns wide. The phase count determines the number of clock cycles to be inserted between each of the outputs. Thus, with a nominal clock rate of 100 MHz, a count of zero results in simultaneous generation of all outputs, while a count of ten results in 100 ns between adjacent output pulses. When the phase is non-zero, the direction bit determines whether a 1..10 or a 10..1 output pulse sequence is generated. Each of the outputs is fed into a high power driver before being sent to the transmit crystal. The location of the board in the bus system is solely responsible for determining which of the twenty phased array outputs the board generates. Thus, the two boards may be freely interchanged.

The high power driver uses a power MOSFET to generate a high voltage pulse. The basic circuit is shown in Appendix C. The principle of the operation of the circuit is described as follows. Assuming that the MOSFET is initially off, capacitor C is charged to the supply voltage Vs through resistances Rs and R. When a gate signal is applied, the MOSFET turns on. Storage capacitor C then discharges through the MOSFET and the resistance R and a pulse voltage appears at the output of the load resistance and transducers.

2.2.2 Receiver Subsystem

The operation of each receiver card is very simple, five control lines feed one or none of sixteen analog inputs to the output. An impedance matching input buffer is located close to the entry point of each signal onto the board. By utilizing surface mount chips it was possible to design the board to process eight signals per side. This pinout provided a simplified analog bus wiring scheme, with each side of a card processing a single row of the phased array. Additionally, by providing redundant signals it was possible to design the board so it could be reversed in the socket without any effect. The location of the board in the bus system is solely responsible for determining which of the sixteen phased array inputs the board processes. Thus, the five boards may be freely interchanged.

2.2.3 IBM I/O Interface

To provide control signals to the phased array a simple interface board based on the IBM-Protoboard was developed. The board is addressed via an I/O port in the PC and provides control, data, and power to the phased array assembly via a DB-25 connector.

2.3 Mechanical Design

A 3-dimensional view of the aluminum array enclosure is shown in Figure 2.7. It supports the head of the phased array and houses the circuit boards, with a DB-25 connector at the rear. Figures 2.8 (a) and (b) show the top and bottom of the case respectively. The sectional view of the phased array assembly is shown in Figure 2.9. Five receiver cards are placed side by side and held in position by the connectors at each end on the two bus boards. A transmitter card is mounted on each side of the receiver cards at right angles to them. Figure 2.10 shows the detail of mounting the head of the phased array.

2.4 Computer and Interfacing

An IEEE488 bus is used to connect an IBM-PC to a CAMAC crate through a LeCroy 8901 interface module. Figure 2.11 shows the overall system layout. The crate houses a LeCroy TR8818/MM8105 transient recorder. Phased array and transmitter control is effected through an eight bit I/O port located on the IBM prototyping board of Section 2.2.3.

The demultiplexed signal is then passed through an external amplifier and filter before being sent to the LeCroy transient recorder for digitization. The digitizer operates at a 100 MHz sampling rate, thus acquiring 20 eight bit samples per wavelength at 5 MHz. Assuming the phased array will operate into steel with a c_p of 6 Km/sec,

$$\frac{6 \text{ km/sec}}{100 \text{ MHz}} = 6 \text{ cm/Ksample} \quad \text{(travel time)}$$

this yields a figure of 3 cm/Ksample for a reflected signal. Thus, a 4K block will permit capture of reflections from up to 12 cm in depth.

2.5 Control and Data Acquisition Software

The control and data acquisition software is written in Turbo Pascal and makes use of assembly language drivers to interface to the IEEE 488 bus. On startup, the software initializes the IEEE 488 port and brings the CAMAC bus to a known state. The digitizer attached to the CAMAC bus is then programmed with its operational values. Additionally, some simple diagnostics on the attached hardware is performed to verify at least partial operation of the system.

Upon completion of these initialization tasks, the user is greeted with the following message:

```
NDT Phased Array,
Data Acquisition subsystem          Version X.XX
Programmed for Dr. H.W. Jones by T.E. Hayman,
on Arctec Canada Limited Subcontract                   Date
```

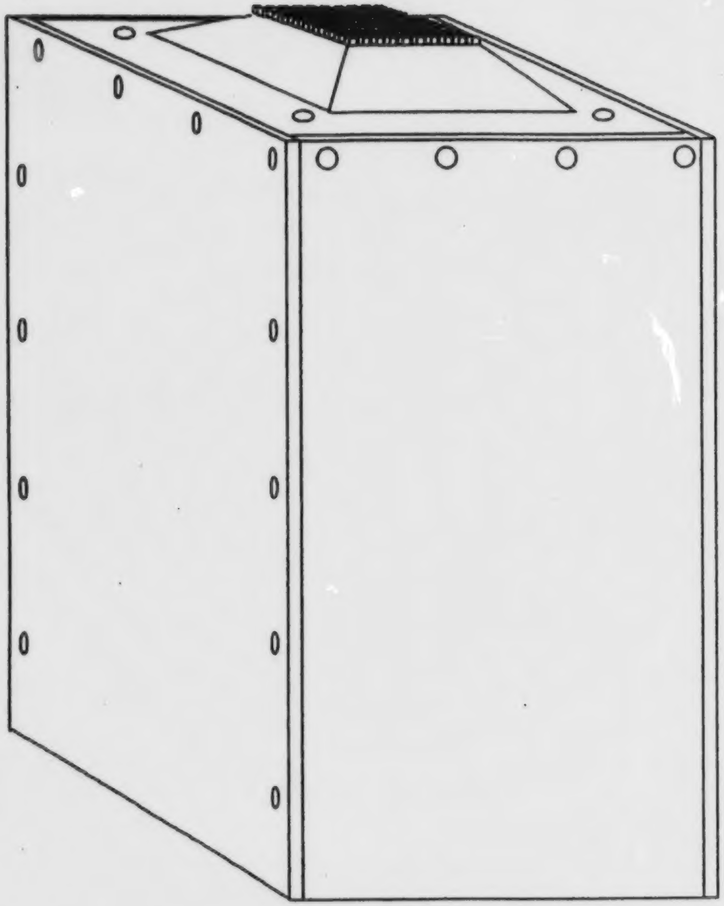
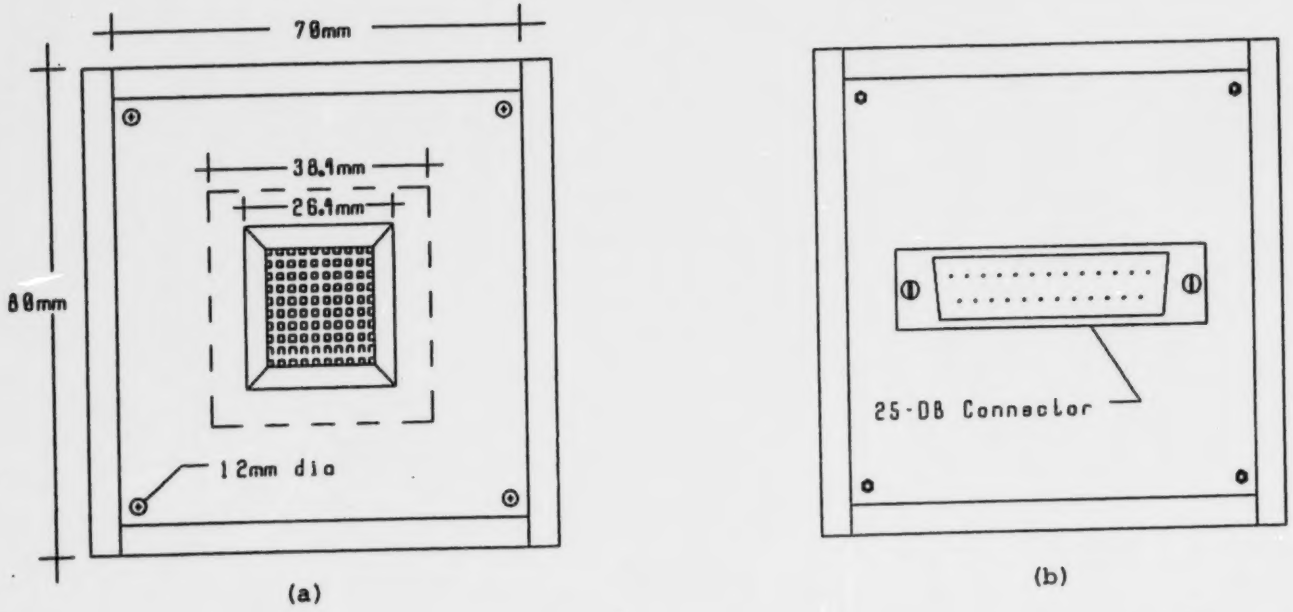


Figure 2.7 General view of the phased array assembly.

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpn	Phased Array Assembly
date Aug 1986	3-Dimensional Model
revised	sheet 1 of 7



70mm
80mm
38.4mm
26.4mm
12mm dia
25-DB Connector

(a)

(b)

Figure 2.8 (a) & (b). Top and bottom view of the phased array assembly.

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpa	Phased Array Assembly
date Aug 1986	Top & Bottom View
revised	sheet 7 of 7

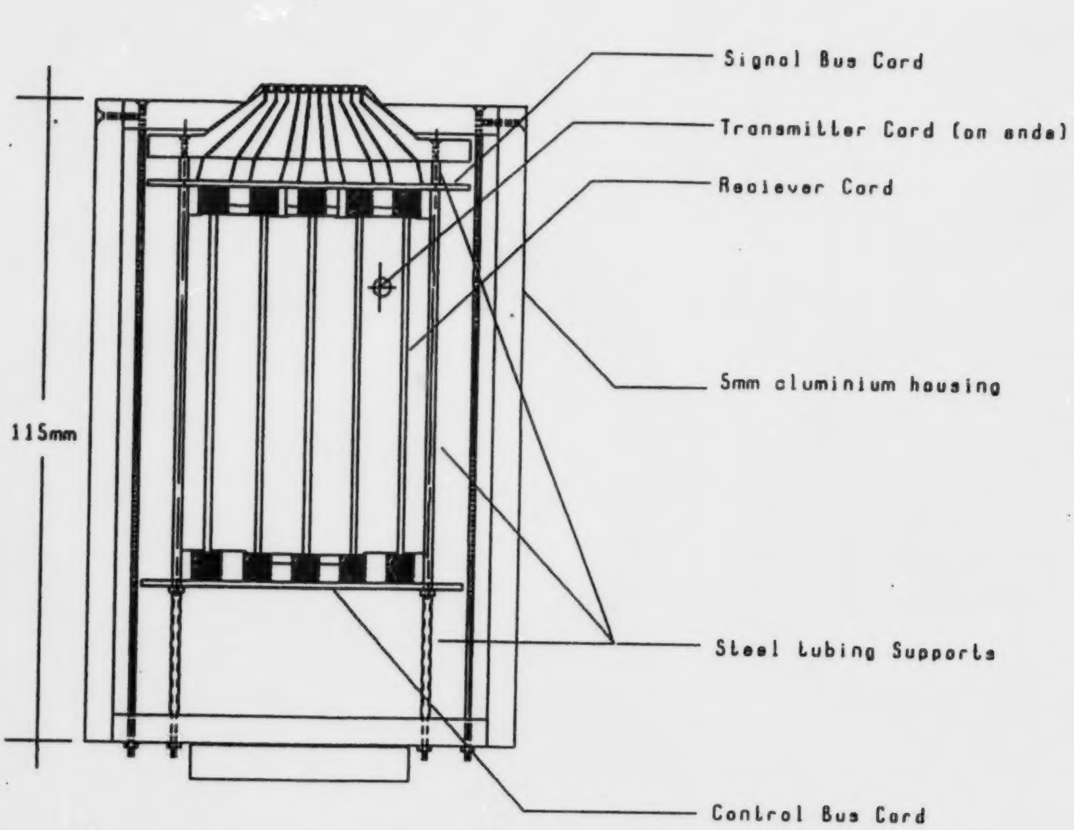

Signal Bus Card
Transmitter Card (on ends)
Reciever Card
5mm aluminium housing
Steel tubing Supports
Control Bus Card
115mm

Figure 2.9 Sectional view of the phased array assembly.

HUGH W JONES & ASSOCIATES LTD.	
designed HWK-TEH	NDT CONTRACT
drawn kpn	Phased Array Assembly
date Aug 1986	Sectional View
revised	sheet 3 of 7

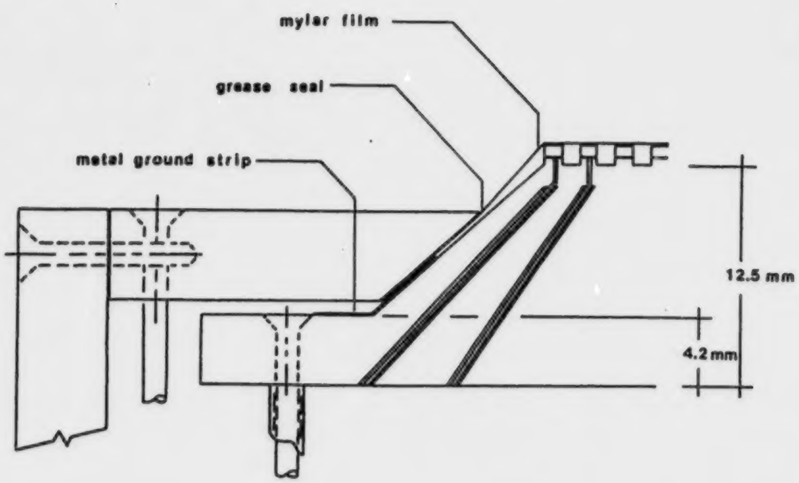
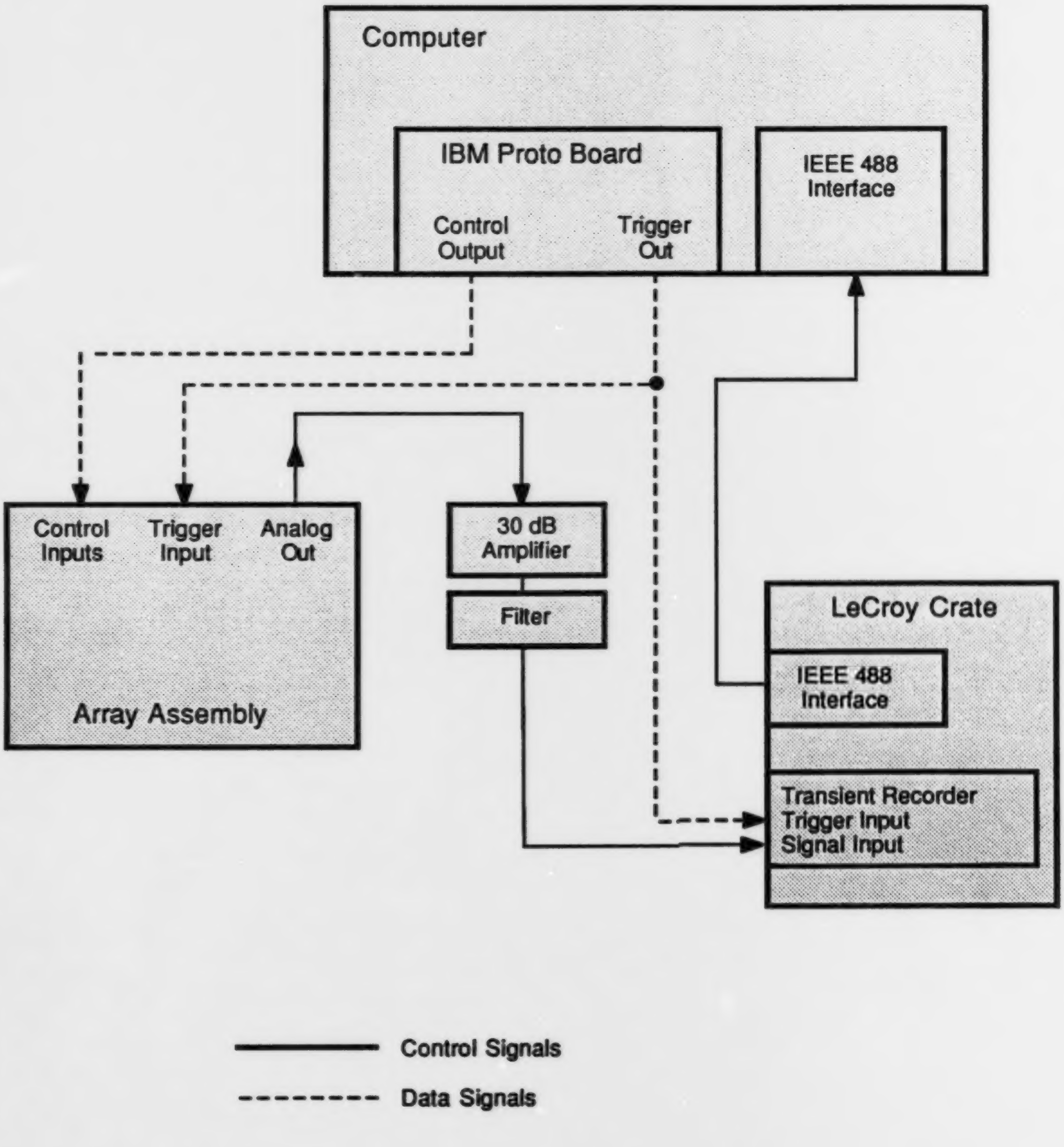

mylar film
grease seal
metal ground strip
12.5 mm
4.2mm

Figure 2.10 Detailed structure of the phased array head.

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpn	Phased Array Assenbly
date Aug 1986	Detail 2 (see page 3)
revised	sheet 5 of 7

Figure 2.11 OVERALL SYSTEM


Computer
IBM Proto Board
Control Output
Trigger Out
IEEE 488 Interface
Control Inputs
Trigger Input
Analog Out
Array Assembly
30 dB Amplifier
Filter
LeCroy Crate
IEEE 488 Interface
Transient Recorder
Trigger Input
Signal Input
Control Signals
Data Signals

After a few seconds, the initial message is replaced by the following prompt:

```
Tx direction [i,m]:
```

i = Angle of rotation, parallel to the array surface
m = Angle off normal to the array surface (see Figure 2.12)

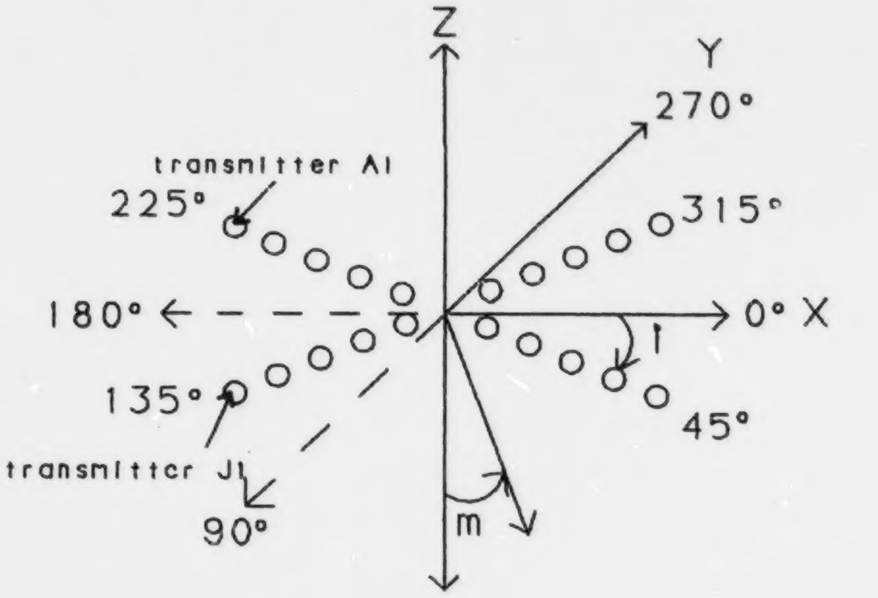
The i parameter is used to determine which transmit lines are to be used and in which direction the lines are to operate. If i equals 0°, 90°, 180° or 270°, both phased array Tx lines will be used. If i equals 45°, 135°, 225°, or 315°, a single Tx line will be used. All other angles are rejected and result in the user being prompted to re-enter the value. The 'm' parameter is used to calculate the time delay between individual transmitter elements within a given line. The time delays calculated will be rounded off to the nearest whole integral of TDU's (Time Delay Unit). For example, with TDU = 10ns (100 MHz.), an 'm' of 0.5° may yield a time delay of 4 ns which would result in the phased array being programmed for 10 ns delays. After determining the angular values, the results are displayed on the console and the following message appears:

```
Acquiring 80 sets of data,
Please stand by...
```

With this message, the software begins the acquisition of 80 individual sets of data into memory. Currently each data set consists of 4 thousand samples of eight bit data. Upon completion the data is written to disk with the following message:

```
Writing to disk...
```

Each data set written to disk has a header associated with it to identify important information for later retrieval. This consists of the date, time, angle for each data set taken, and any other pertinent information as may be dictated at a later date. Upon completion of the acquisition, the first prompt reappears and the process may be repeated for each desired angle. A list of the software is described in Appendix D.


Z
Y
270°
transmitter A1
225°
315°
180°
0° X
l
135°
45°
transmitter J1
90°
m

Note: Transmitters are on the X-Y plane

Figure 2.12 Geometry of the transmitted beam parameters.

3. SYSTEM CONSTRUCTION AND ASSEMBLY

3.1 Construction of Transducer Head

The transducer head is the heart of the device and its production is one of the most difficult of the "mechanical" tasks.

Based on the designs produced in June 1986, the head consists of one hundred ceramic elements, each only 1 mm square, backed by a support molded epoxy loaded with tungsten powder. Connection wires and metallic shields had to be fastened to each and fed to the electronics.

From the start it was known that this would not be easy to fabricate but it was certainly not expected to be as difficult as it turned out to be. At each stage in the manufacturing process, new techniques had to be created and tried. Each stage presented problems that had not been obvious one stage earlier, making it necessary to repeat work already considered complete.

The unit was built in the following steps:

1. Preparation of mold and pre-insertion of wires.
2. Molding of base.
3. Attachment of ceramic in one large piece.
4. Cutting ceramic into 1 mm squares.

In the following paragraphs the various stages will be described and the problems and solutions of each presented.

3.1.1 Preparation of the Mold

Firstly, a mold was prepared in six pieces; each side being separately made so that the final part could be easily removed. Then a grid of one hundred small holes was drilled in the top and the bottom parts. This had to be carefully and accurately done because the final connections to the ceramic pieces would rely on the spacing between these holes. A tolerance of .1mm was requested in both x and y dimensions on the top or "front" piece (i.e. the one to hold the ceramic).

After the mold was assembled, one hundred very fine #34 wires were strung through from front to back through the two sets of holes. Then the needles were removed from one hundred medical syringes and individually cut and ground to length. This was not so easy because the cutting process naturally closed the hole in the needle and each had to be individually cleaned out with a special tool. The edges of each tiny hole needed to be free from burrs or rough edges because any sharp edge would cut the fragile insulation and short out the element.

The heads also had to be carefully treated as the wires, once molded in, could not be replaced and were very prone to breakage at the surface of the hard epoxy. As well the wire insulation could still be damaged if it was allowed to rub against the sides of the needles.

3.1.2 Molding of Base

After the mold was prepared it was fixed in a horizontal position ready for pouring. The epoxy was prepared by mixing the two parts in the correct proportion and then adding tungsten power. This made a very thick material and so the mold was heated by an incandescent light bulb and vibrated during the pour to prevent the formation of voids between the wires. While hardening, the mold was kept at about 60°C to speed the process, which took about 36 hours to complete.

When hard the mold side were disassembled and carefully broken free from the base. Next the top and bottom were removed - again a difficult process because the wires, being fed through the many small holes, were holding the parts together. In order to remove them each wire had to be carefully eased back through the two plates, without damaging it or, by forcing the plate too hard, any other wire. The head of the base was then ground flat, removing excess wire from the front of the head and exposing a grid of one hundred small copper dots, i.e. the ends of the wires showing where they had been cut off by the grinding process. Lastly, conductive epoxy was applied carefully on the back of the head to connect the ends of the needles as shields. It was then allowed to set for 24 hours.

3.1.3 Attachment of Ceramic

Prior to the cutting of the ceramic plate into small squares it had to be fastened securely onto the head with a conductive cement. This did not prove to be easy and the method has not yet been perfected. When the ceramic is cut, it is subjected to quite severe stresses, especially during the final stages of the prosess and the glue used to hold it on is severely tested. Unfortunately failure does not appear until the whole head is nearly completed and it negates all the work previously put into the manufacture.

After several tries the following steps were found to be best:

1. Clean head and ceramic with "Clean-sol", a commercial preparation used to remove traces of silicone and then with a "Freon" based cleaner to remove the "Clean-sol".

2. Spread a thin layer of freshly mixed conductive epoxy onto the head and gently press the ceramic down onto it. If this is done too hard there is a chance of the ceramic breaking and also a chance of spreading the epoxy so thin that it does not adhere properly.

3. Leave the head and ceramic at a temperature of 60°C for at least 48 hours to cure well.

When the project began, it was intended to add a thin layer of brass to the top of the head, over the ceramic, to improve the acoustic coupling to steel. This proved impossible within the limitations of expenditure and time. To do so one would have to fasten the brass over the ceramic with glue and then cut through both brass and ceramic when cutting the small square element grid. it was tried a number of times but every time the brass would come off during the cutting process. Eventually it was decided to concentrate on just cutting the ceramic alone, without worrying about the brass.

3.1.4 Cutting Ceramic

As was expected this caused the greatest problems. At first it was felt that the use of a diamond cutter traversed slowly across the ceramic ten times in each direction would be the best method. A suitable diamond saw was procured in the USA and imported, reaching Canada in December 1986.

Tests were then made with trial pieces of ceramic fastened to epoxy. Unfortunately the saw moving across the part, however carefully done, caused the edges of the ceramic to chip and the circular motion tended to lift the tiny ceramic squares off the base. Another problem was the difficulty in making ten cuts at precise distances in two directions, although if the cuts had been more successful a jig might well have been built.

The only alternative which appeared viable by late January was that of cutting with fine steel wires and grinding paste, so a special machine was lashed up to test the idea. Ten 1 mm piano wires were tensioned in a frame and the center to center spacing was set at 2 mm. A motor from an automobile windshield wiper was fastened to it so that it pulled the frame back and forth across the ceramic. The system worked well and did not chip the ceramic but suffered from a lack of rigidity.

By February, a satisfactory frame had been developed. To keep the wires in place as they cut, a finger was attached to the base of the head which fitted up between two of the cutting wires, keeping them straight (See Figure 3.1).

Throughout the whole process of developing a system for cutting by wires, the bonding of the epoxy holding the ceramic was a problem. Initially it was thought that the straight cut of the wires would exert no upwards force on the ceramic pieces but either this is not true or some other force is coming into play because even then some of the small ceramic squares came off at the end of the cut. Their numbers were so few that they could be glued back on individually.

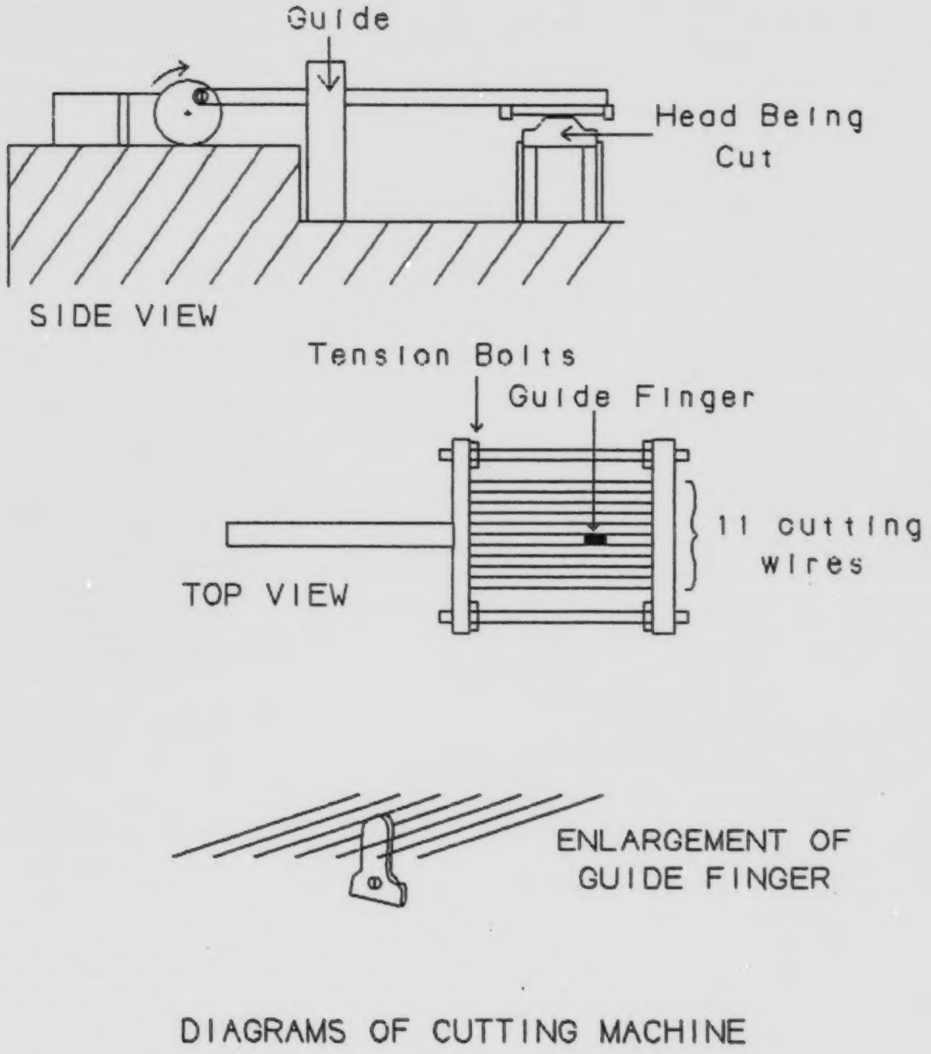

Guide
Head Being Cut
SIDE VIEW
Tension Bolts
Guide Finger
11 cutting wires
TOP VIEW
ENLARGEMENT OF GUIDE FINGER
DIAGRAMS OF CUTTING MACHINE

Figure 3.1 Diagram of crystal cutting machine.

There was a thought that the grease base of the cutting paste might be chemically attacking the joint but various different bases made no difference. Water was finally selected due to its easier cleaning.

3.2 Construction of Electronic Components

As for the transducer head, the electronic parts offered challenges of their own. In particular the packaging of the various circuits into the required space was a major consideration, as was the high (100 MHz) clock speed.

All of the electronics were built with discrete components. Because of size restriction it was decided to use as many as possible of the new surface mount components.

Twenty driver amplifiers for the transmitting elements were needed and eighty preamplifiers for the receivers. Each driver required logic circuits to select and time its output, and the outputs from the receivers had to be controlled and multiplexed for the LeCroy digitizer.

Ten circuit boards were used, using surface mount components on both sides. Two were identical transmitter boards housing ten drivers each with their logic. Five were receiver boards with sixteen preamplifiers each, together with their multiplexers. Of the last three, two were "mother boards" joining the rest and making connections and the last was an extra board with the 100 MHz clock circuitry.

3.2.1 Preamplifier Boards

There are two major design sections; the preamplifiers themselves and the multiplexers. Each will be covered separately below:

3.2.1.1 Preamplifiers

The main considerations, apart from small size, were high input impedance and wide bandwidth, centred at 5MHz. It was estimated that the capacitance of the small ceramic sections would be of the order of 10 to 20pF and thus to avoid low frequency roll-off the input impedance had to be in the order of 100 Kohms or greater. To give this level a VHF junction FET was selected, the U310, which is conveniently available in surface mount form. A simple design of source follower was adopted, both to conserve space and to provide the lowest possible output impedance. On each board sixteen of these were mounted on the "front" with their outputs led to the multiplexing circuits.

3.2.1.2 Multiplexers

A CMOS type 4051 multiplexer was chosen as the best compromise available given the bandwidth, analogue signal capability and surface mount requirements. It has the required size and signal handling capability but its bandwidth is marginal at 10MHz. Wiring was kept short and separate to avoid cross-coupling between channels.

3.2.2 Transmitter Driver Cards

The transmitter driver boards consist of a complex high speed TTL switching system utilizing 74F74 integrated circuitry to provide the twenty trigger pulses (100nsec wide) for the next stage. Each of these outputs is fed into a high power MOSFET driver. These drivers consist of step up transformers which provide higher voltage outputs and matching to the transmitting crystals. It must be emphasized that the requirements of 10nsec are certainly state-of-the-art.

3.2.3 Oscillator Card

It was originally intended to have this circuit on the Control Bus Card but there was no room and an extra 100MHz oscillator card had to be added.

3.2.4 Control Bus Card

This card contains 8 connectors on a 60mm square card together with some electronics. It was decided to put the electronics on the underside of the card between the connectors, a tight squeeze resulting in many jumpers.

The original connectors were not available except on 14 week delivery so substitutes had to be found. The complexity of construction necessitated plated feed through holes for the header sockets. Rigiflex of Halifax was subcontracted to produce the control bus cards. After a period of time the boards were received and it was found that the connector pins would not fit through the plated holes. Due to time limitations a new board could not be manufactured, so surface mount connectors were substituted. In the process of reflow soldering, the surface mount connectors suffered some heat damage. The damaged connectors seemed operable and time limitations forced the use of these connectors on the final unit.

3.2.5 Assembly of Electronic Card

The cards were put together much as previously planned. Five receiver cards are placed side by side and held in position by the connectors at each end on the two "bus boards". Two transmitter cards flank the receivers at right angles to them, forming a square. The oscillator card has been added to the "back" of the unit, parallel to the bus card. Interconnections between the various cards are by plug and sockets and to the unit as a whole by a DB-25 connector at the rear of the unit.

3.3 Mechanical Construction

The circuit boards connected together are supported by an aluminum case. The head is attached to a cover plate which also supports the bus card. The wires from the head of the array assembly run to the appropriate connectors on the analog bus card.

Over the head a film of aluminum foil is held against the ceramic elements by a polyethylene cover. This provides a ground return for the elements. Due to the larger transmitter boards, the overall size of the unit is longer than originally designed.

3.4 Summary of Construction Procedures

1. Construct Head:

   a. Form mold
   b. Feed wires through mold
   c. Prepare syringe needles
   d. Feed wires through needles
   e. Pour epoxy, let set 24 hours at 60°C
   f. Grind off excess wire on front of head
   g. Clean ceramic and head
   h. Glue ceramic with conductive adhesive (36 hrs)
   i. Put conductive adhesive on rear of head, let set
   j. Cut squares with wire machine and grinding paste
   k. Rotate 90° and repeat
   l. Clean head and glue on any squares which have come off

2. Construct Electronics:

   a. Construct circuit boards
   b. Test boards
   c. Attach head wires to bus board
   d. Attach oscillator card to bus board
   e. Attach and wire up output connector
   f. Assemble cards together

3. Assemble mechanical and electrical parts.

3.5 Problems Encountered

These have been described in the individual sections.

4. TESTING AND RESULTS

The testing process was performed in the two distinct areas of electronic and acoustic performances. A complete evaluation of the electronics driving the acoustic system was made both to confirm its performance and to identify any weaknesses in the prototype design.

The acoustic performance tests were made for the same reasons as well as to provide data that might be required for the Phase II image reconstruction procedure.

4.1 Test Procedures

Initial testing of the system revealed a number of deficiencies which had to be dealt with before formal testing could be undertaken. A majority of these problems were minor in nature, primarily construction errors, which were quickly identified and corrected. One problem however, resulted in a relaxation of the original design specifications. The initial design made use of a 100MHz reference clock to generate 10 nsec phase increments for the transmitter system. When operating into steel, this was to provide linear steering between 0° and 25° in 1.7° increments. During initial evaluation of the system, it became clear that the phase system was inoperative for frequencies above 30MHz. The problem was traced to a modification that Orion Electronics Ltd. had implemented during the construction stage to circumnavigate another problem which they had encountered. Our attempts to correct the situation without major revisions were not entirely successful. The circuit could be made to operate reliably with a clock frequency of 60MHz, but the design frequency of 100MHz remained elusive. At the higher frequency, the operation of the system was not reliable across all the possible delays, in particular those with minimum phase. As the cases of minimum delay were felt to be critical for evaluation of the phased transmitter, a reference clock of 60MHz was utilized. This reduction of clock frequency resulted in a degradation of the original design goals by increasing the minimum step angle of the steering system. For example, the operation into steel was now between 0° and 48.6° in 2.8° increments. As these angles were deemed to be acceptable for the purposes of initial testing, the time constraints of the testing schedule dictated that the higher operating frequencies be abandoned.

4.1.1 Electronics

Seven distinct subsystems were identified and targeted for tests, they were:

1) Oscillator
2) Control bus
3) Transmitter phase control system
4) Transmitter pulse generators

5) Transmitter high power drivers
6) Receiver Preamplifiers
7) Receiver multiplexing system

The control bus and oscillator tests involved basic tests to verify signal generation. The input signals were fed into the system and the generation of the required signals at each connector were verified.

Under control from the computer, the generation of the phased control signals for the complete range of programmable phases were verified. Additionally, the timing of the signals was measured to ensure accurate phase generation. The signals for both of the transmitter boards were measured concurrently to ensure coherent phase between the independent circuits.

Once the operation of the phased control pulses was verified, the generation of a 1/2 cycle, 5 MHz pulse for each phase line was verified. This signal was then traced to each of the twenty power drivers on the two boards. The amplitude and phase of the output wave form for each channel was measured as a function of the programmed phase.

The 80 receiving preamplifiers were checked by feeding a small 5 MHz signal from a function generator to the input of preamplifiers. The receiver multiplexing system was also verified concurrently.

4.1.2 Acoustic Aspect

The acoustic performance of the array was evaluated using a steel test specimen and a miniature transducer, both made specifically for checking the directive pattern. Ideally the steel specimen should have been hemispherical but the manufacturing costs were prohibitive. Instead, three cylindrical rods (O.D. 6", lengths = 4", 6", 9.5") were made polished end faces. (All reported results were obtained with the 4" long sample.) Two miniature transducers were built with the same structure as the phased array, i.e. 5MHz PZT crystals with tungsten loaded epoxy on the rear faces. See Figure 4.1.

The phased array assembly was attached to one side and held in place by three supporting rods and a spring loaded plate. A grid of 0.1 inch squares was placed on the other side of the test sample to act as a reference for the measurements. Figure 4.2 shows the phased array attached to the test sample.

The transmitting transducers were checked for proper operation. Two problems were found. First, five transmitting transducers were not working properly, i.e. A10, D4, H8, and J1. This is probably due to a bad contact of the electrode on the rear face of the crystal. To fix this problem is not an easy task (it would require a new phased array head). The quick solution is to disconnect the wire from the driver to the transmitting transducer and use the other 15 transmitting transducers. Second, the output of the drivers were not strong enough to penetrate the 4" steel sample.

Figure 4.1

Picture of two miniature transducers.

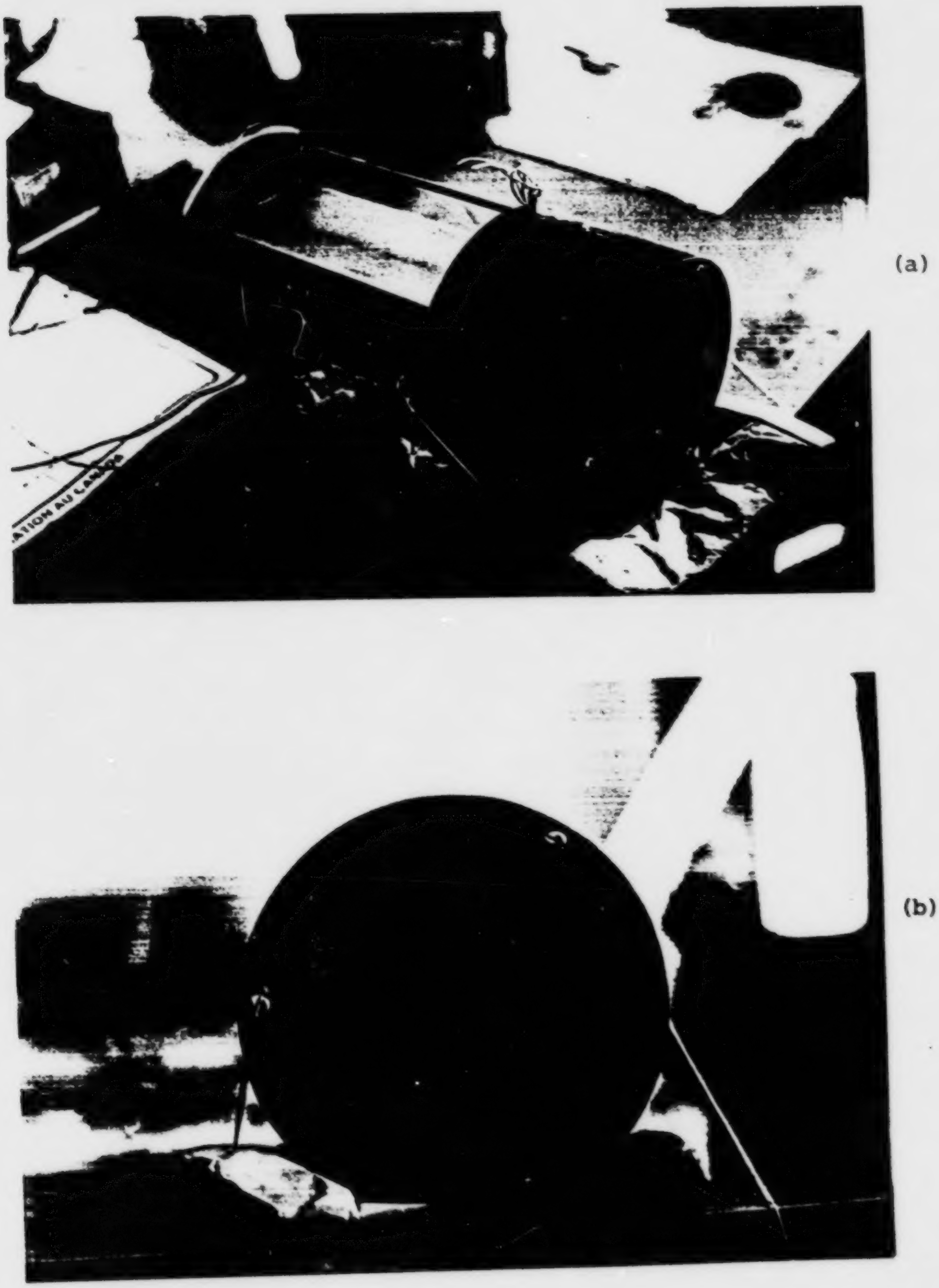


Figure 4.2

Experimental Set up

A driver (IC 7407) and a pull-up resistor (470 ohms) were added to the driver board before the gate signal into the power FET, in order to boost the gate signal and obtain higher output from the FET. The revised transmitter schematic is shown in Appendix C. The additional components were mounted on two small circuit cards, each of which was wired to one of the original transmitter cards. Fortunately the new cards were small enough so as to fit inside the enclosure. With the new circuitry the output level of the transmitting transducers varied from 40V to 80V as shown in Table 4.1. The different output levels of the transmitting transducers will increase slightly the side lobe of the directive pattern.

Ten receiving elements have problems as indicated by their electrical characteristics in Table 4.2, zero or low capacitance indicating a short circuit, probably between the connecting wires and the shielding hypodermic needles.

Another problem is that the crystals pulled off during the cutting process were re-glued. This would not be a problem except for the additional thickness of cement left some crystals raised above the rest so that the remainder could not make good acoustical contact with the test specimen.

4.2 Beam Characteristics and Assessment of Performance

The miniature transducer is used to receive the signal transmitted through the test sample. This signal is fed into an amplifier and filter, then the result is displayed on the oscilloscope for observation and measurements. By placing the transducer at each grid point, the directive pattern of the transmitter array will be mapped. The same procedure is repeated for each selected delay to obtain the directive patterns.

To demonstrate the functioning of the transmitter we present the results of the tests which were performed using the set-up shown in Figure 4.2. The tests were:

1. The spatial plot of the maxima of the sound pulse envelope for different steering angles shown in Figures 4.3 to 4.6. Maximum delay is 48.6° off center, i.e. 4.5" off center for 4" long block. Therefore the maximum steering angle cannot be checked on the present testing block.

2. The shape of the resulting pulse for conditions stated in Figure 4.7.

There is a considerable amount of scatter among the points measured in Fig. 4.3. It is probably due, at least partially, to variations between the transmit element outputs, less than ideal acoustic coupling due to the unevenness of the array surface, as well as measurement error. The possibility of some defect in the test cylinder also exists.

TABLE 4.1

TRANSMITTER OUTPUT
(supply voltage 85V)

CRYSTAL	OUTPUT VOLTAGE (V)	CRYSTAL	OUTPUT VOLTAGE (V)
J1	---	A1	44
I2	64	B2	76
H3	84	C3	74
G4	90	D4	---
F5	76	E5	64
E6	82	F6	82
D7	52	G7	72
C8	---	H8	---
B9	54	19	74
A10	---	J10	36

TABLE 4.2

DEFECTIVE RECEIVERS

CRYSTALS KNOWN TO HAVE DEFECTS:

CRYSTAL	RESISTANCE (ohms to gnd)	CAPACITANCE (pf)	RECEIVED AMPLITUDE (V)
A5	86.6	---	---
E7	182.0	---	0.61
J2	8.9	---	---
J3	4.8	---	---
J6	12.2	---	0.43

CRYSTALS WITH UNSTABLE MEASUREMENTS:

These crystals have readings that change frequently. The measurements listed are valid for the time of testing only.

CRYSTAL	RESISTANCE (ohms to gnd)	CAPACITANCE (pf)	RECEIVED AMPLITUDE (V)
A3	---	38.0	0.52
C1	---	----	0.55
G9	130k	28.2	0.60
H1	120k	38.1	----
J5	30	----	0.70

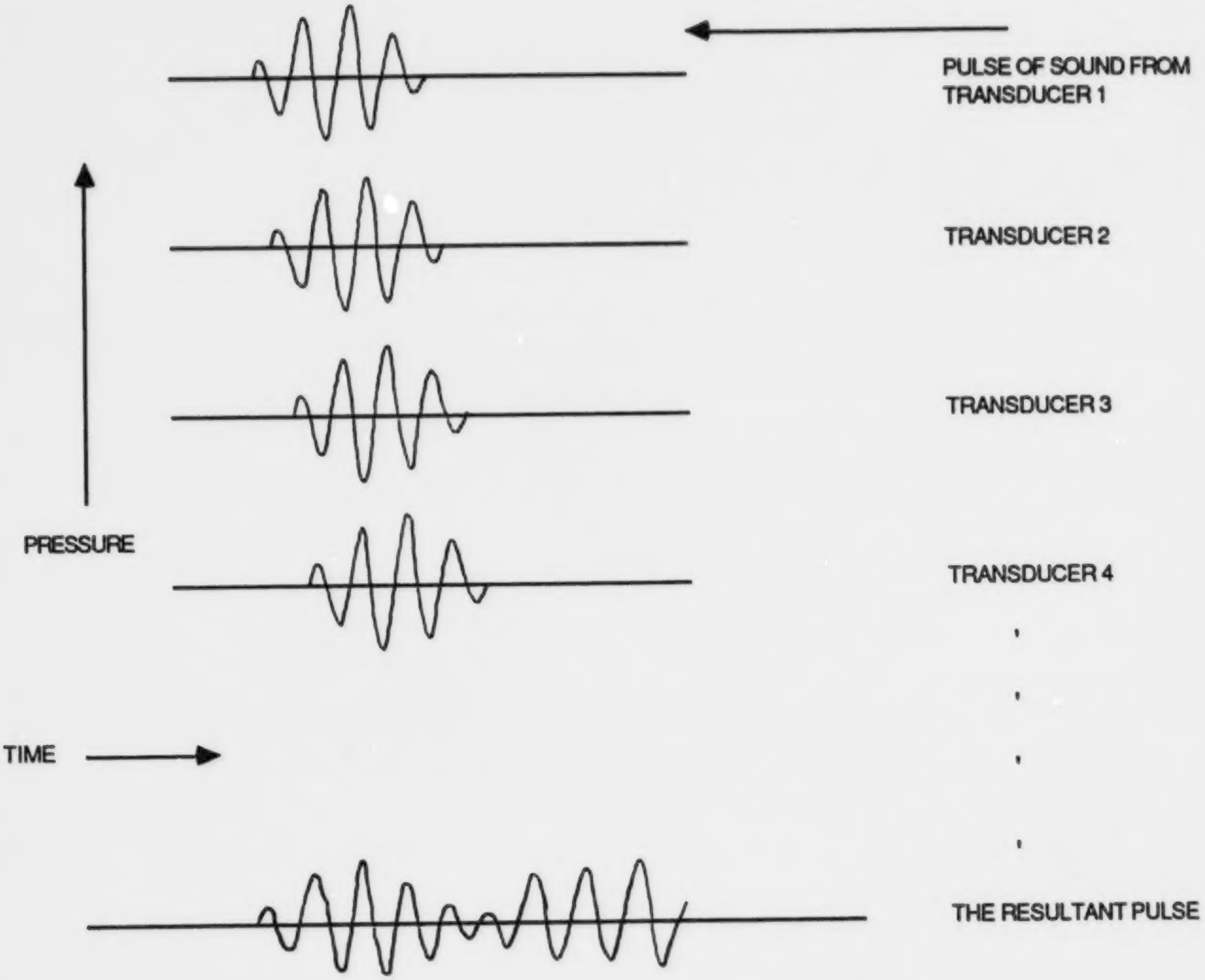

PULSE OF SOUND FROM TRANSDUCER 1
TRANSDUCER 2
TRANSDUCER 3
PRESSURE
TRANSDUCER 4
TIME
THE RESULTANT PULSE

FIGURE 4.3

THE RESULTANT PULSE IS THE SCALE SUM OF THE PULSES FROM MANY TRANSDUCERS

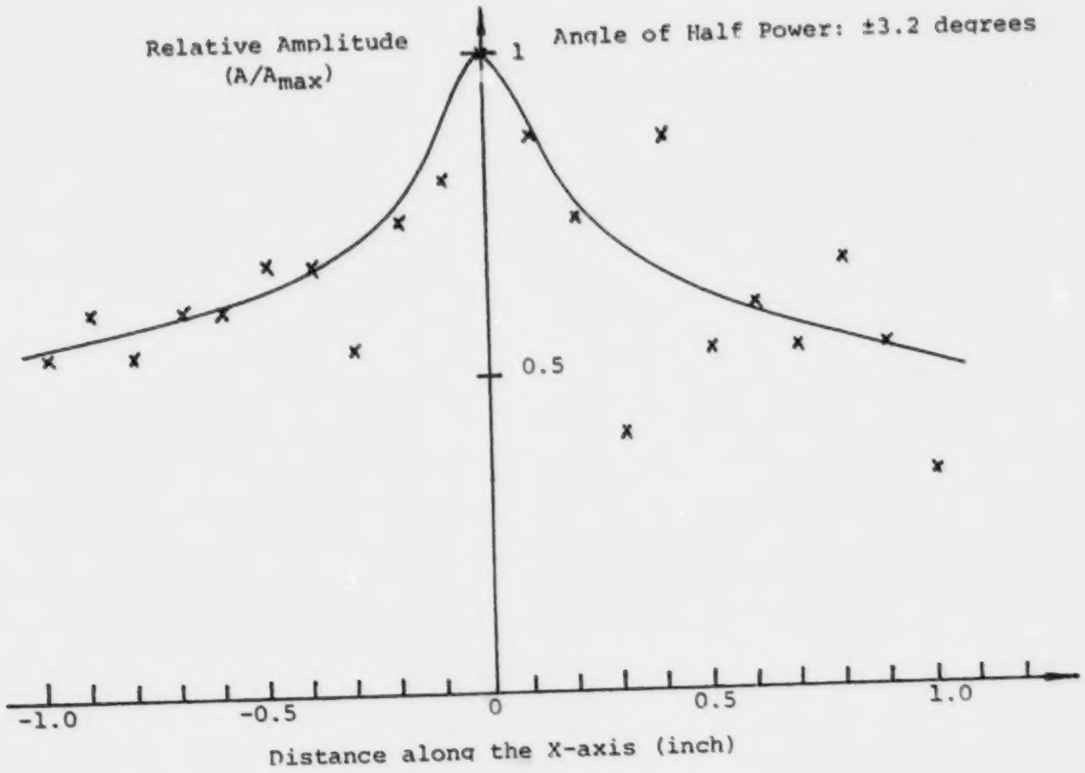


Figure 4.4

The directive pattern of the transmitters with no delay on X-axis.

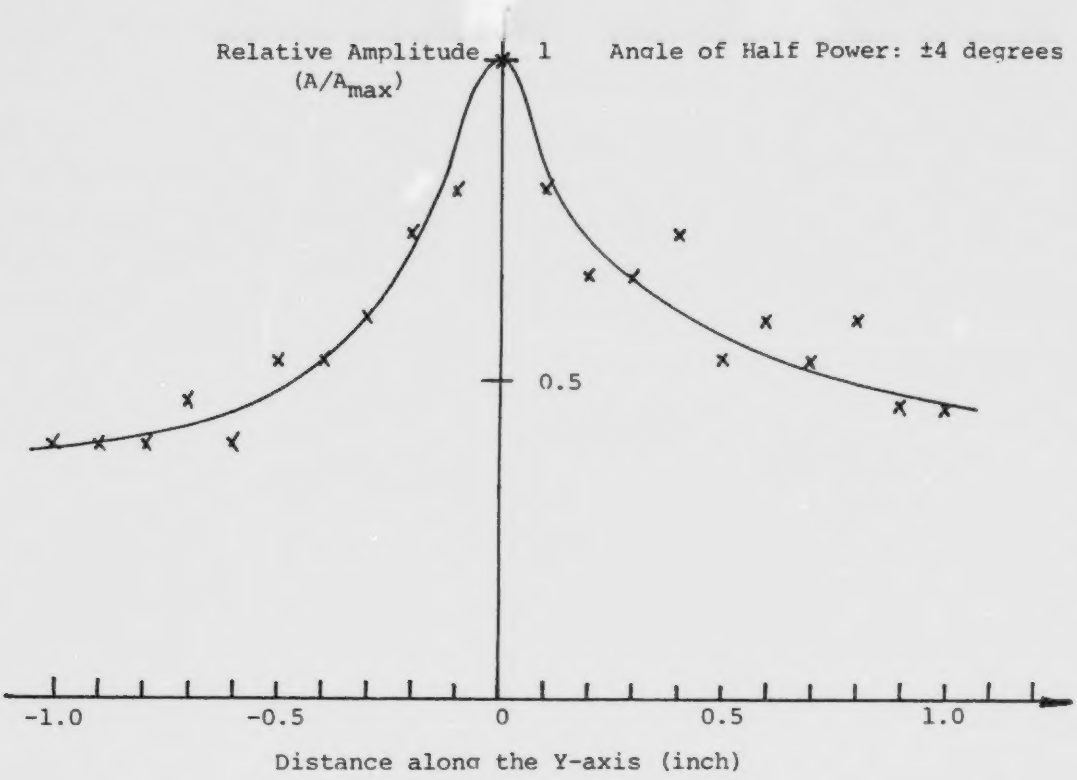


Figure 4.5

The directive pattern of the transmitters with no delay on Y-axis.

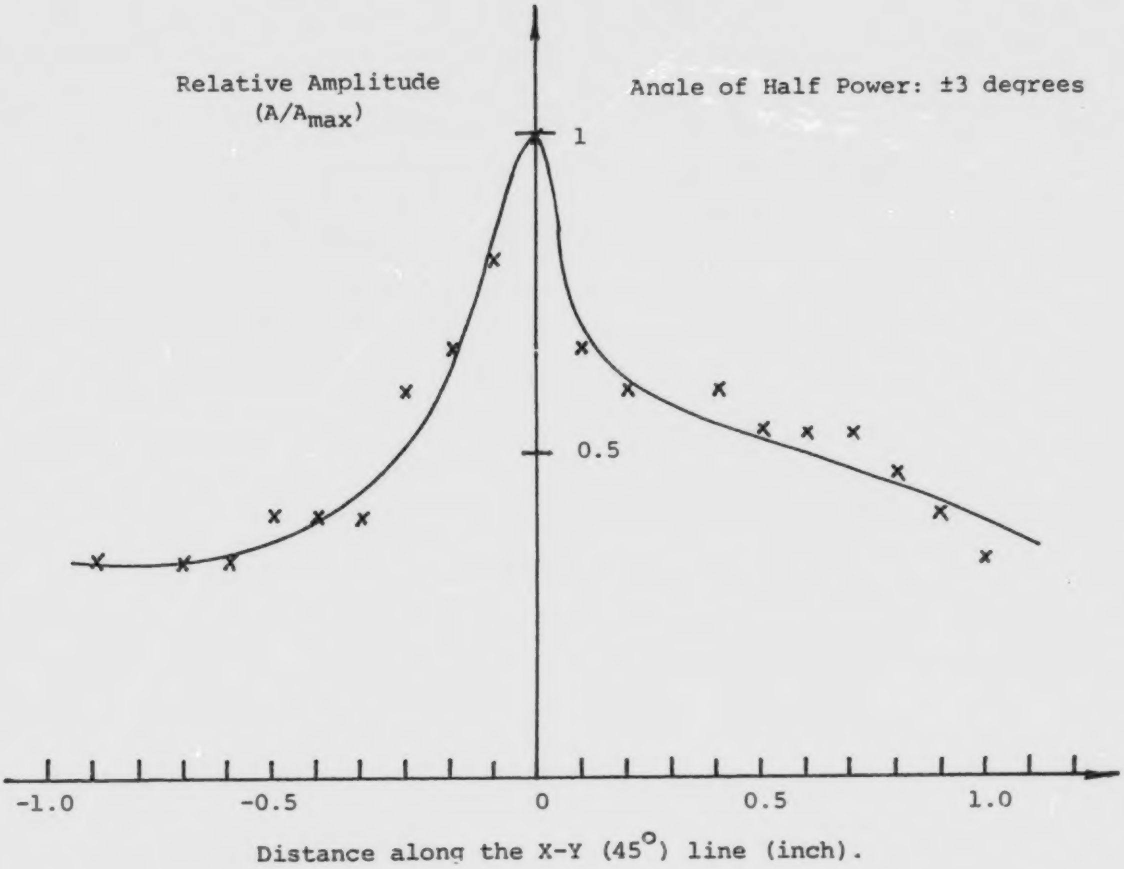
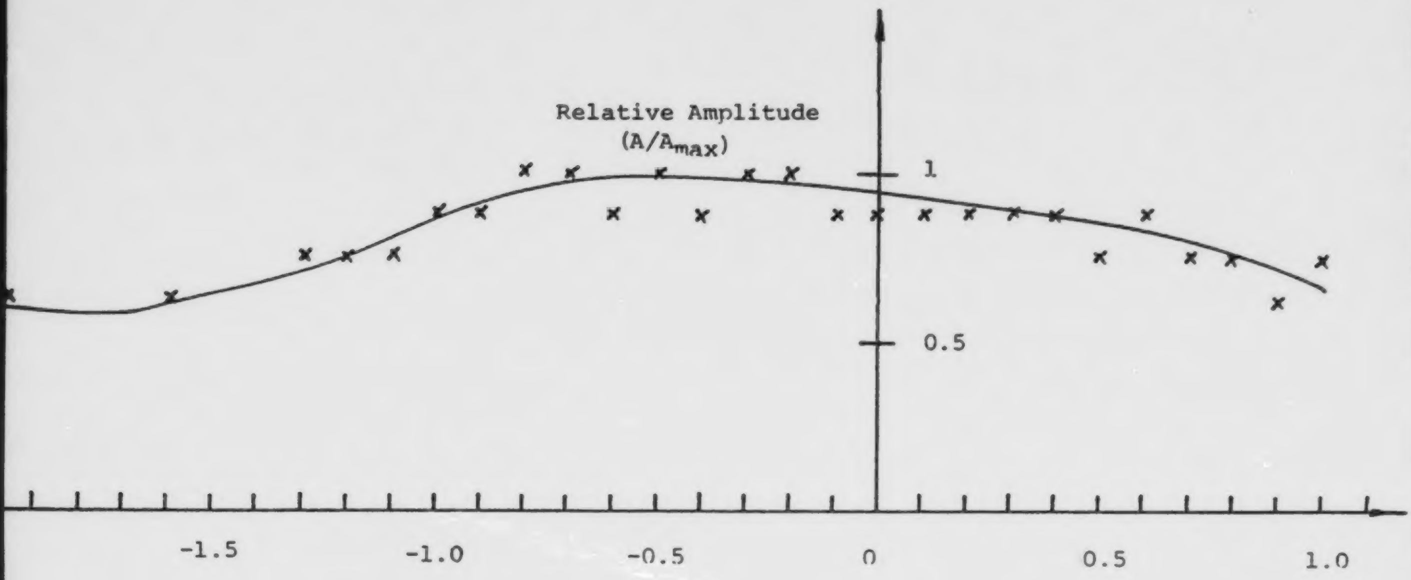


Figure 4.6

The directional pattern of the transmitters with no delay on X-Y (45°) line.



Distance along the Y-axis (inch).

Figure 4.7

The directional pattern of the transmitters with delay 5 on the Y-axis. (delay 5: 14.5 degrees off the center)

The clearly defined peak in Fig. 4.3.-4.7 is not as sharp in Fig. 4.6. It cannot be readily compared to the previously calculated continuous waveform results because of the different way in which pulsed waveforms combine.

We prepared reports earlier on beam patterns to be expected from continuous wave transmitting transducers (this was an extra to the contract and a copy of that report is attached hereto). Unfortunately it is not possible to test for the CW focusing pattern directly, nor does it seem advisable to make such a determination unless it is specifically called for (the enclosed reference discusses the point in principle). The problem can be seen by reference to Figure 4.8. As the length of the pulse increases the shape of the resulting pattern, which results from the sum, changes. This change is dependent on position, pulse length and relative amplitude of the pulses. One can imagine that as the peak of one wavefront is reaching a measurement point, the peak of another wavefront has already passed (or is yet to arrive). Thus even if the wavefronts are in phase, the resulting combination will differ from the CW case. This effect becomes more pronounced as the differences in distance between the source elements and the measurement point are increased, as is the case in Fig. 4.6. In principle, the situation does not have very considerable consequences because the computer will reconstruct with time gating, using the sound field at a specified instant of time for the image reconstruction.

We are not satisfied with the output driving voltage of the electronic drivers to the piezoelectric elements. They are below that which was specified in our design. We accept that O'Rion Electronics Limited have put their best effort into the problem and we cannot reasonably expect more from them given the funding provided. We know that they went very considerably over cost in making the system that we have now got. We shall endeavour to improve the situation by our efforts, if need be to re-design and perhaps make a new transmitter board. The receiving circuitry is satisfactory but the element unevenness is a major problem.

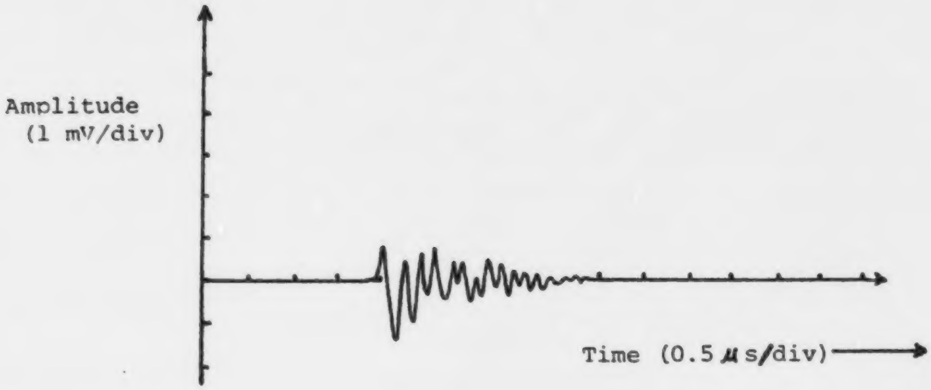


Figure 4.8

The resulting pulse of 10 transmitting pulses measured at the center of the test specimen.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Phase I saw the design and construction of a 100 element two dimensional acoustic array, the associated electronics and a suitable interface to an IBM compatible computer. Among other requirements, the electronics demanded high precision timing for proper phasing of the transmit elements, and the use of sub-miniature components to achieve a reasonably sized system. To this end the work was most successful. The electronics met most design specifications except the system clock frequency was reduced from 100 MHz to 60 MHz to provide more stable operation, variations between the driver output voltages remains a minor problem, and the receiver gains are lower than desired.

The acoustic design of the array elements was relatively straightforward given the desired beam characteristics and the ubiquitous practical constraints. The major bottlenecks were related to the actual construction, in particular the problems of getting good connections between the PZT crystal and the #34 wires without any inadvertent ground to the hypodermic needles, and the tendency of the brass and PZT ceramic to come unglued during the cutting of the bulk material into array elements. The wiring problems were for the most part overcome by experience and attention to detail. An acceptable cutting procedure for the PZT has yet to be proven although it is felt the original diamond saw can work.

5.2 Recommendations

As with any prototype system, there is always room for improvement. The following areas are felt to need further work:

1. The transmit drivers need design changes to reduce variations among their output voltages and improve overall efficiency.

2. The ability to individually fire selected transmit elements should be incoporated. This would be a great help in verifying proper system operation in normal use, as well as in the case of trouble. For instance, looking at received reflections with single transmit elements fired would help identify any transmit element in poor contact with a test specimen, or problems with one of the electronic channels.

3. At present an external 30dB amplifier and low pass filter are used between the array electronics and the LeCroy digitizer. Incorporating these within the array housing would help simplify the system and improve its immunity to noise.

4. The FET receiver amplifiers should be changed so as to have a voltage gain of about 4 rather than just buffering the signal with a gain of less than 1, as is the case now.

5. Another head should be built, utilizing the experience gained in Phase I, with the intention of perfecting a crystal cutting procedure.

6. Ultimately, it would be very useful to have the actual transducer array at the end of an umbilical cord linking it to the electronics, making the system much more practical to use. It would involve designing a mount for the array backing material and a one hundred wire cable with suitable impedance characteristics.

APPENDIX A

TRANSDUCER DESIGN

Appendix A

Transducer Design

The piezoelectric disc model of KLM is shown in Figure A1. In this model, the piezoelectric disc is represented by an acoustic transmission line tapped at its center, and driven by a perfect transformer of ratio $\phi:1$, where

$$\phi = k_T \, (\pi/\omega_o c_o z_c)^{\frac{1}{2}} \operatorname{sinc} \left(\frac{\omega}{2\omega_o}\right) \tag{1}$$

with sinc $X = \sin(\pi X)/\pi X$. The transformer has a capacity in series with it of value $c_o = \epsilon^S A/\ell$, the clamped capacitance of the transducer. There is an additional reactance in series with the transformer which can be modeled as a variable capacitance of value:

$$c' = -c_o/k_T^2 \operatorname{sinc} (\omega/\omega_o) \tag{2}$$

which is negligible near resonance. ϵ^S is the electrical permitivity at constant strain. A is the surface area of the transducer, ℓ the thickness of the ceramic, and $\omega_o = \pi \nu/\ell$, ν = velocity in ceramic and k_T is the effective piezoelectric coupling coefficient.

The transducer can be considered as a set of finite or infinite transmission lines in which a number of plane acoustic waves are running in both directions. Therefore, a schematic of the ocmplete transducer as a series of cascaded two-ports is shown in Figure A2. The values of matrices $N_1 \ldots\ldots\ldots N_8$ are given in the Appendix A1. The results of the calculations can be summarized by means of the model in Figure A3. This model has three transfer matrices which represent:

i) an electrical port
ii) the backing plus the left half of the transducer
iii) the right half of the transducer plus the front load ($\lambda/4$ layer(s) and ice), respectively. The overall transfer matrix can be written

$$N_{tot} = N_f N_m N_e = N_8 N_7 N_6 N_4 N_3 N_2 \tag{3}$$

If we take the electric impedance at the electric port Z_e and the load at the front acoustic port Z_T, we get from

$$\begin{pmatrix} \bar{F}_t \\ \bar{V}_t \end{pmatrix} = \begin{pmatrix} N_{t11} & N_{t12} \\ N_{t21} & N_{t22} \end{pmatrix} \cdot \begin{pmatrix} \bar{F}_i \\ \bar{V}_i \end{pmatrix} \tag{4}$$

where $\overline{F_t}$, $\overline{F_{t'}}$ generalized voltage, $\overline{V_t}$, $\overline{V_{t'}}$ generalized current and all variables are given in the Laplace transformed way.

the transfer function of transmission

$$H_t\ (P) = 2Z_t/(-Z_eZ_tN_{t21} + Z_eN_{t11} - N_{t12} + Z_tN_{t22}) \qquad (5)$$

From the above discussion, if the excitation pulse is known, then the transmitted signals can be calculated.

Appendix A1

$$N_1 = \begin{pmatrix} n & -\frac{PL}{n} \\ 0 & \frac{1}{n} \end{pmatrix}$$

$$N_2 = \begin{pmatrix} 1 & -\frac{C^0 + C'}{PC_0C'} \\ 0 & 1 \end{pmatrix}$$

$$N_3 = \begin{pmatrix} \frac{1}{\phi} & 0 \\ 0 & \phi \end{pmatrix}$$

$$N_4 = \begin{pmatrix} 1 & 0 \\ -\frac{1}{Z_b'} & 1 \end{pmatrix} \qquad Z_b' = Z_c \cdot \frac{Z_c \sinh(P\tau) + Z_b \cosh(P\tau)}{Z_c \cosh(P\tau) + Z_b \sinh(P\tau)}$$

$$N_5 = \begin{pmatrix} \cosh(P\tau) & -Z_c \sinh(P\tau) \\ -\frac{1}{Z_c} \sinh(P\tau) & \cosh(P\tau) \end{pmatrix}$$

$$N_6 = \begin{pmatrix} \cosh(P\tau) & -Z_c \sinh(P\tau) \\ -\frac{1}{Z_c} \sinh(P\tau) & \cosh(P\tau) \end{pmatrix}$$

$$N_7 = \begin{pmatrix} \cosh(P\tau_1) & -Z_1 \sinh(P\tau_1) \\ -\frac{1}{Z_1} \sinh(P\tau_1) & \cosh(P\tau_1) \end{pmatrix}$$

$$N_8 = \begin{pmatrix} \cosh(P\tau_2) & -Z_2 \sinh(P\tau_2) \\ -\frac{1}{Z_2} \sinh(P\tau_2) & \cosh(P\tau_2) \end{pmatrix}$$

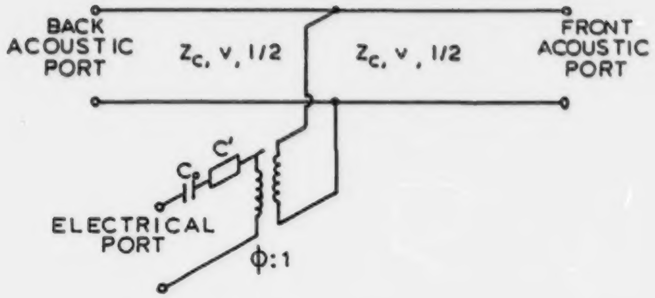

BACK
ACOUSTIC
PORT
Z_c, v, l/2
Z_c, v, l/2
FRONT
ACOUSTIC
PORT
C_0
C'
ELECTRICAL
PORT
φ:1

Figure A-1 The transducer model proposed by Krimholtz et al, the so-called KLM-model.

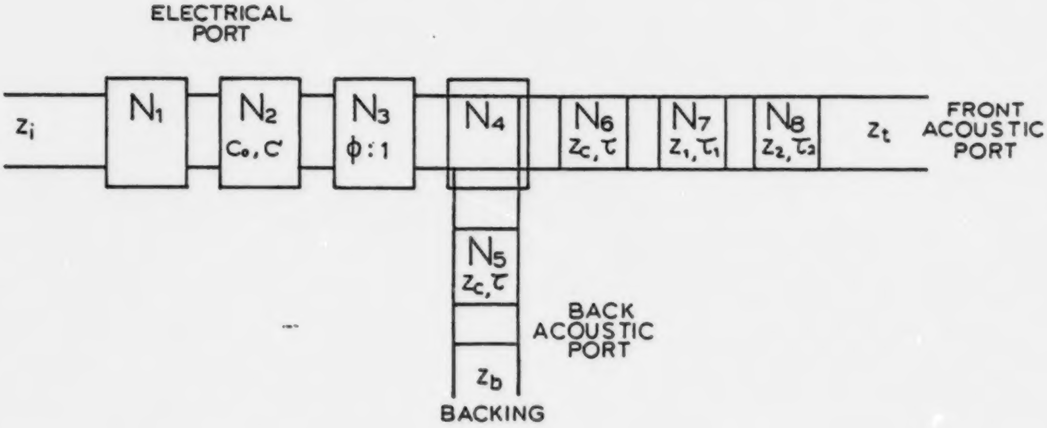

ELECTRICAL
PORT
Z_i
N_1
N_2
C_o, C'
N_3
φ:1
N_4
N_6
Z_c, τ
N_7
Z_1, τ_1
N_8
Z_2, τ_2
Z_t
FRONT
ACOUSTIC
PORT
N_5
Z_c, τ
BACK
ACOUSTIC
PORT
Z_b
BACKING

Figure A-2 The complete transmission-line scheme corresponding to the KLM-model.

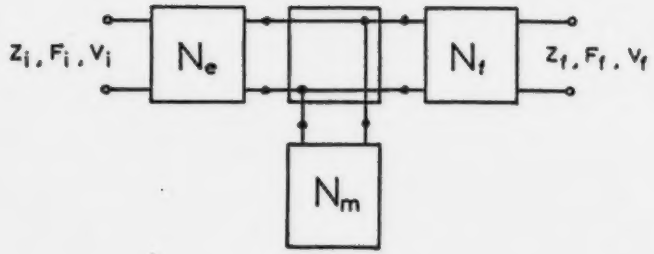

Z_i, F_i, v_i
N_e
N_f
Z_f, F_f, v_f
N_m

Figure A-3 Simplified scheme of the KLM-model.

APPENDIX B

DIRECTIVE PATTERN OF TRANSMISSION ARRAY

1. INTRODUCTION

In an acoustic imaging system, it is usual to use a transmitting array to illuminate an object in order to get a good image. There are two reasons for using a transmitting array:

<a> Improve signal/noise ratio, i.e. minimize acoustic intensity outside the observed range, and maximize the intensity inside of the range;

<b> To illuminate the object as uniformly as possible.

In comparison with the far field, found at long distances from the transmission array, the acoustic field close to the array is rather complicated. In any case, it is preferable to put the object in the far field.

In fact, whenever the object is to be observed it is often put quite near to the acoustic lens in order to improve resolution of the image. Therefore, it is necessary to know the extent of the near field and the intensity distribution on an object at a certain distance. Then it can be known whether it is necessary and possible to compensate for nonuniformity of illumination.

This report discusses problems concerned and some analogous results are given for four types of transmission arrays, including

<1> the extent of near field,
<2> the axial response, and
<3> the intensity distribution patterns for various distances from array.

2. EXTENT OF NEAR FIELD

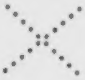
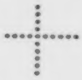
The intensity distribution pattern displays a strong interference effect at the near field. In contrast, the far field shows a simpler beam pattern but, there is no exact demarcation between the far field and the near field.

A criterion for the inner boundary of the far field is that the distances from the various portions of the array to a point on the acoustic axis must not differ by more than about one half-wavelength. This yields that the beginning of the far field is roughly

$$R_{min} = 0.25*L^2/W_L . \qquad \text{Eq.1}$$

where W_L is the wavelength of the acoustic wave and L is the greatest extent of the array. Table.1 gives the extents of near field for four types of transmission arrays, estimated by Eq.1 with wavelength of 1.22 mm.

Table.1 Extent of Near Field (unit: mm)

Array Number	1	2	3	4
Array Structure				
Element Number	20	19	4	8
Greatest Extent of the Array	25.5	18	2.8	8.4
Extent of Near Field	133	66	1.6	14.4

3. AXIAL RESPONSE

The field along the acoustic axis of an array may be used as an indicator for the near field and far field. The axial response at great distance displays a monotonically decreasing behavior going asymptotically to a r^{-1} dependence, i.e. spherical divergence. For small distances, the axial response displays a strong interference effect. Figures 1 to 4 show the axial responses for different arrays. These figures may give more information about near field and far field. Of course, intensity distribution patterns for different distances to the array give the complete information necessary for acoustic imaging engineering.

4. TRANSMISSION PATTERNS

Analogous intensity distribution patterns for different distances to transmission arrays of four types are obtained by using an IBM-PC XT. Figures 5.1 - 8.20 show the results. These figures show the patterns in the range of an object whose image will be formed by the acoustic imaging system. This range was referred to as the observable range in the previous report. The spherical decay r^{-1} has been compensated in the analogue.

Figures 5.1 - 5.6 show the transmission patterns for the distances of 6 mm - 500 mm to a small array, No. 3. Since all these are far field, they are exactly the same after compensation for spherical decay.

Figures 6.1 - 6.20 give an example of transfer from near field to the far field. Although the distances of 15 mm to 30 mm belong to far field according to the extent of the near field in Table.1 estimated by Eq.1, the patterns change remarkably. So the extent of the near field may be more than that estimated by the traditional equation, Eq.1.

Another two sets of transmission patterns; Figures. 7.1 - 7.20 and 8.1 - 8.20, show that large arrays have a main lobe illuminating a very small area in the center of the observed range and the patterns in the near field may change dramatically even if the distance changes by only a few millimeters.

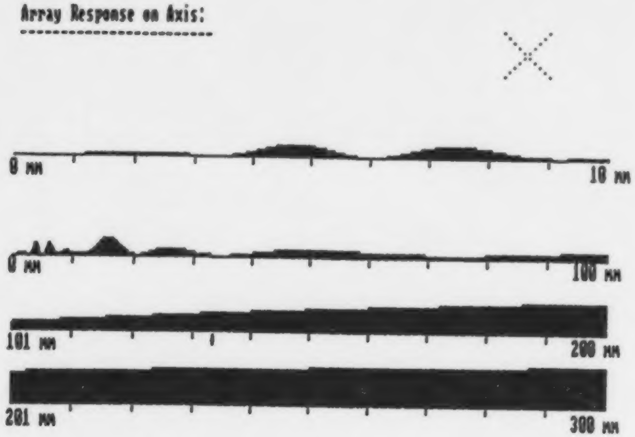


Fig. 1

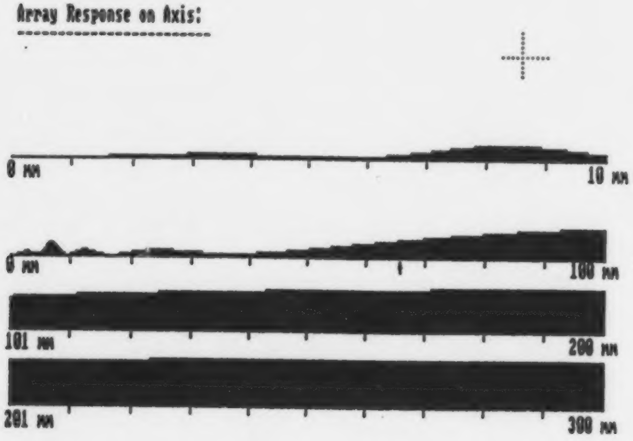


Fig. 2

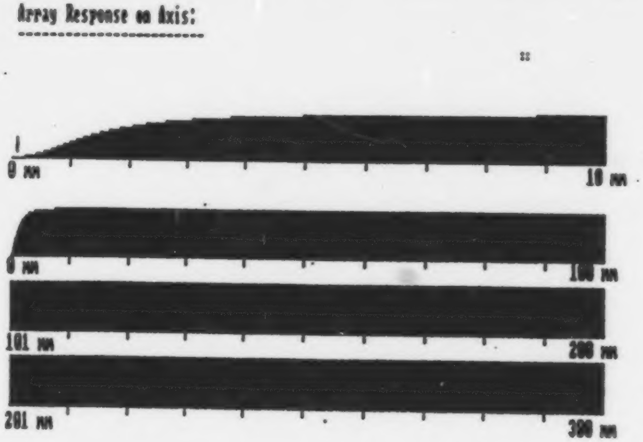


Fig. 3

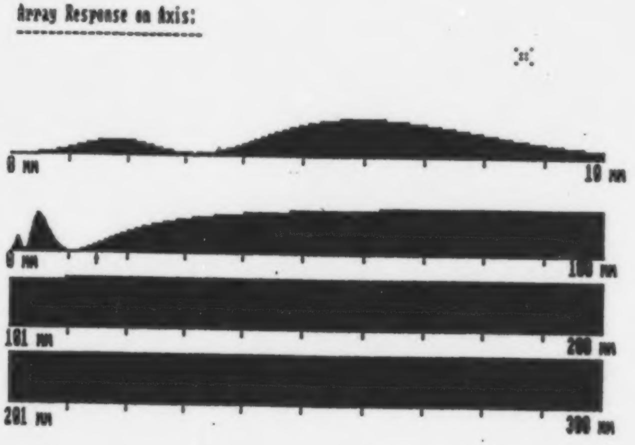


Fig. 4

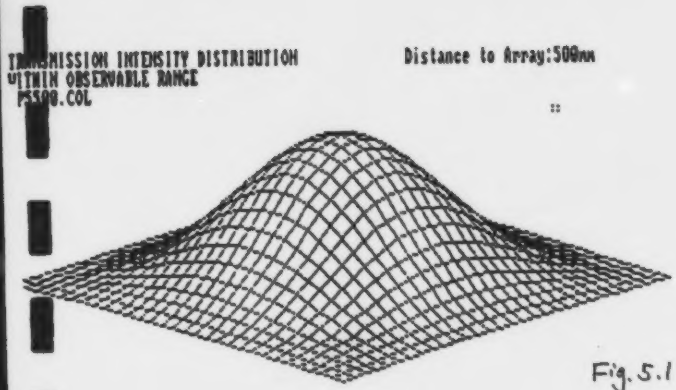


Fig. 5.1

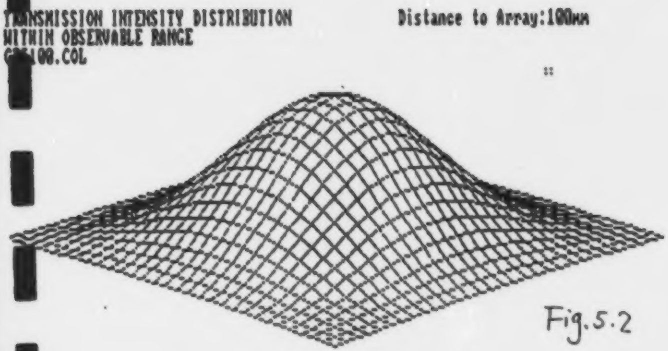


Fig. 5.2

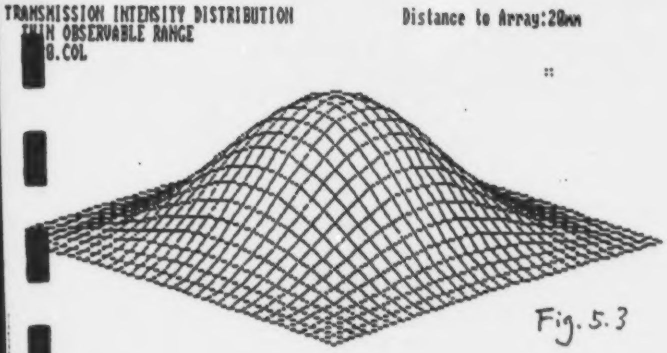


Fig. 5.3

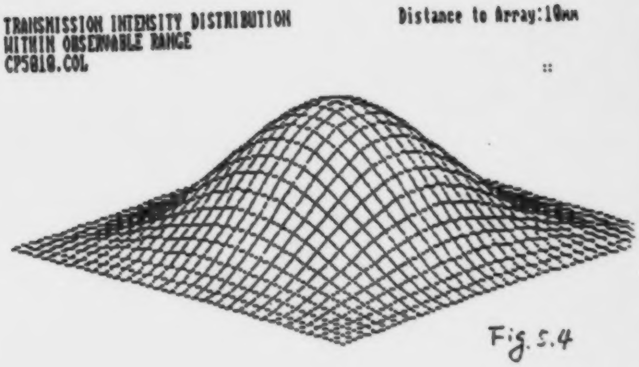


Fig. 5.4

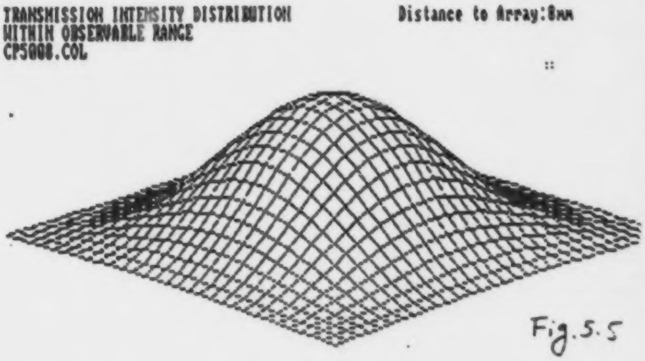


Fig. 5.5

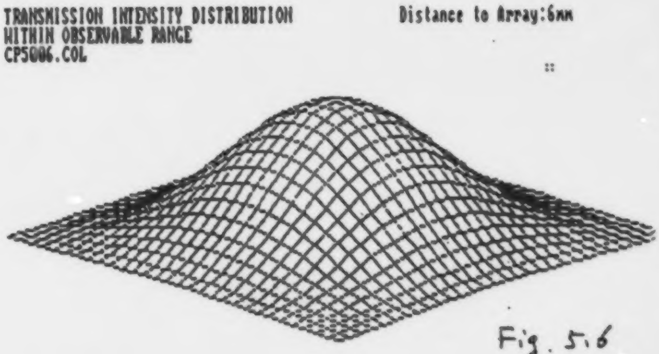


Fig. 5.6

Fig. 5.1 – 5.6

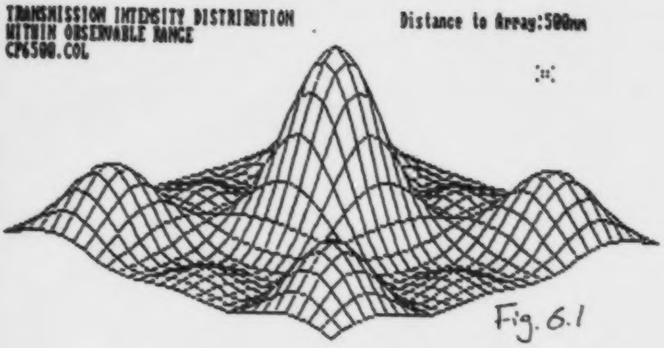


Fig. 6.1

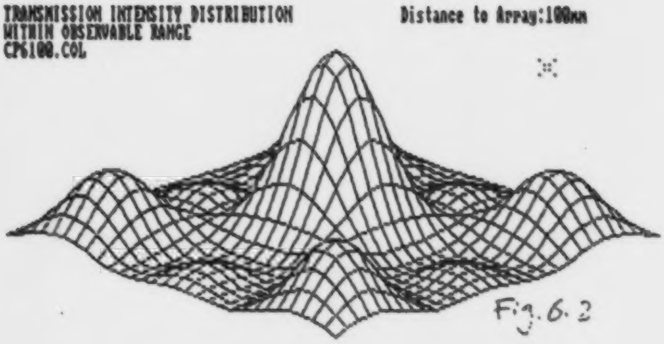


Fig. 6.2

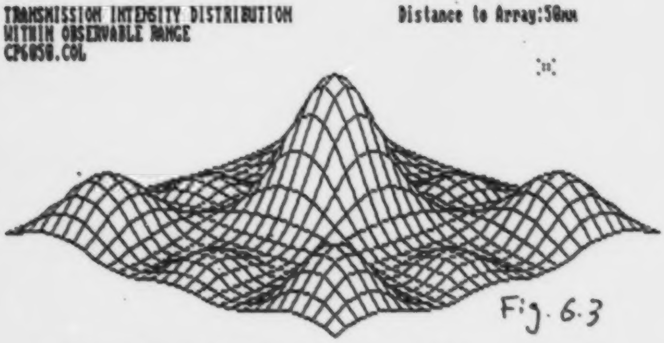


Fig. 6.3

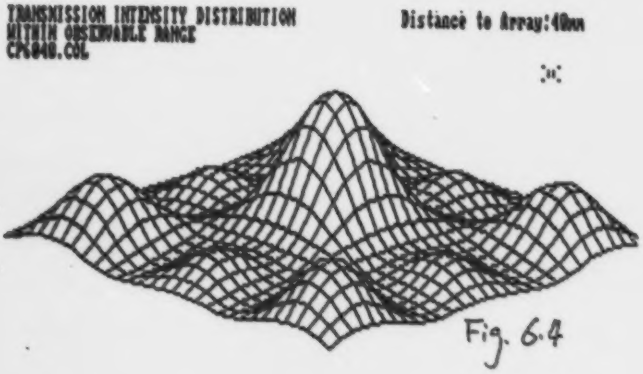


Fig. 6.4

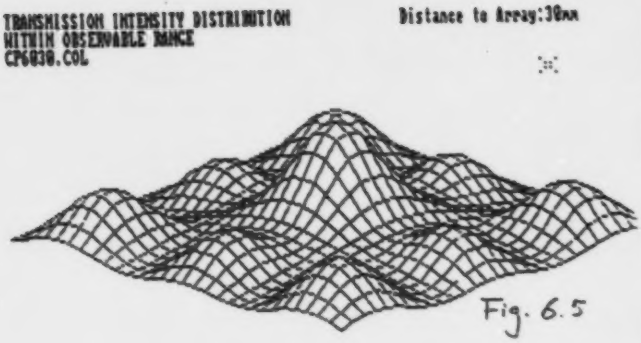


Fig. 6.5

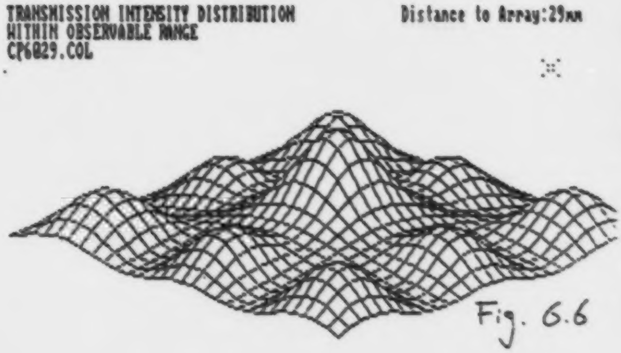


Fig. 6.6

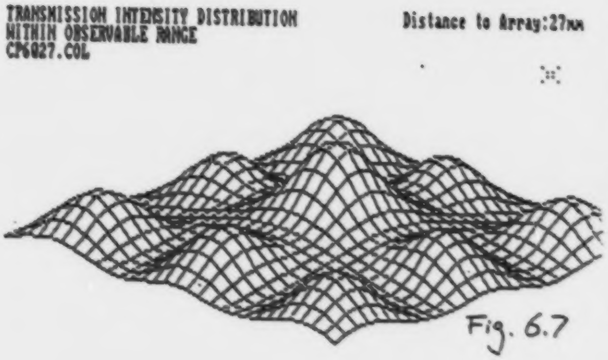


Fig. 6.7

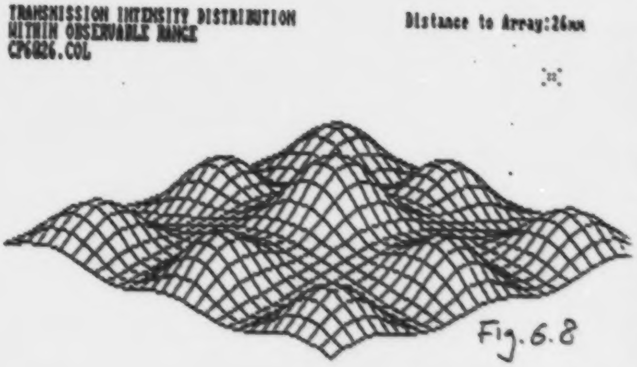


Fig. 6.8

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6025.COL

Distance to Array:25mm

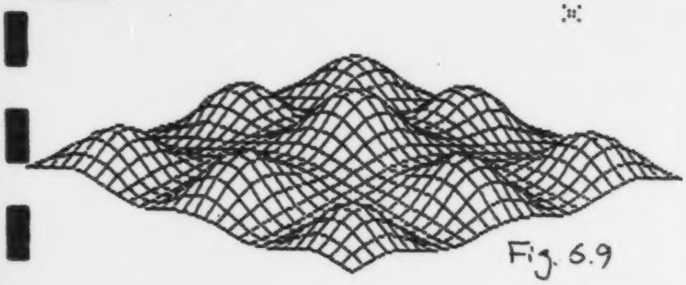


Fig. 6.9

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6024.COL

Distance to Array:24mm

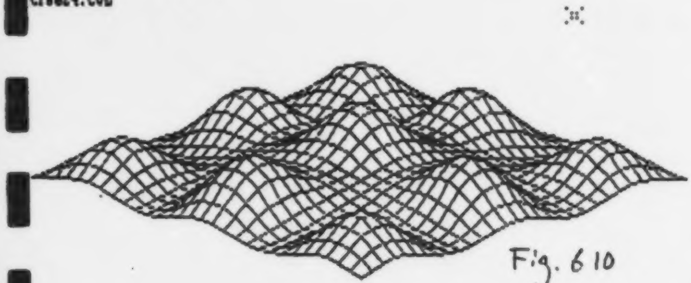


Fig. 6.10

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
P6023.COL

Distance to Array:23mm

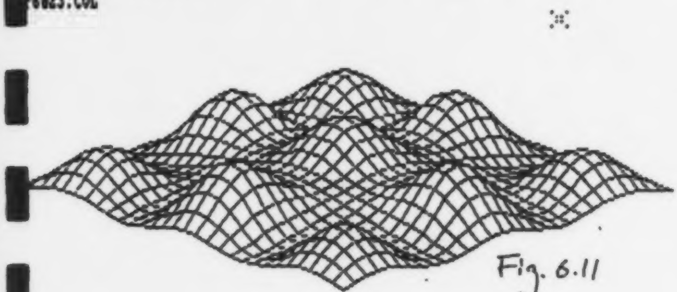


Fig. 6.11

NSMISSION INTENSITY DISTRIBUTION
THIN OBSERVABLE RANGE
CP6022.COL

Distance to Array:22mm

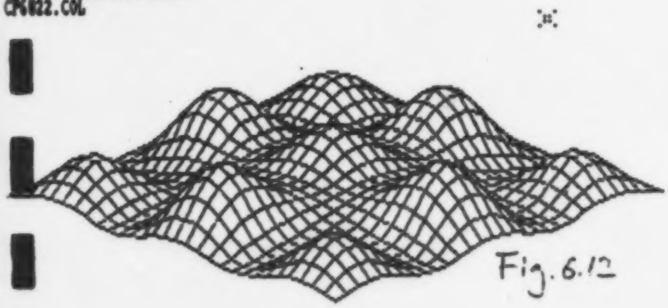


Fig. 6.12

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6021.COL

Distance to Array:21mm

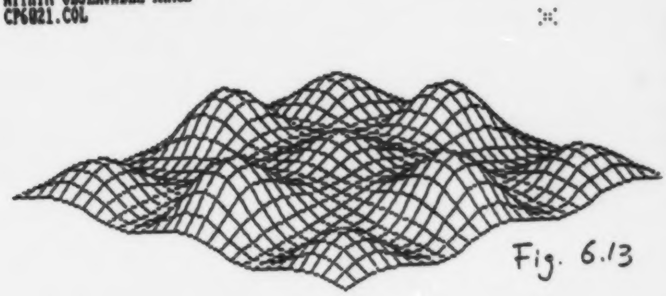


Fig. 6.13

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6020.COL

Distance to Array:20mm

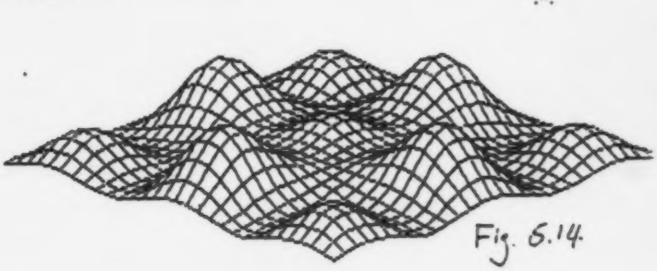


Fig. 6.14

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6019.COL

Distance to Array:19mm

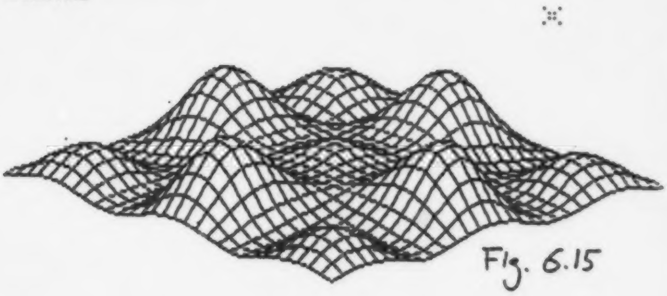


Fig. 6.15

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6018.COL

Distance to Array:18mm

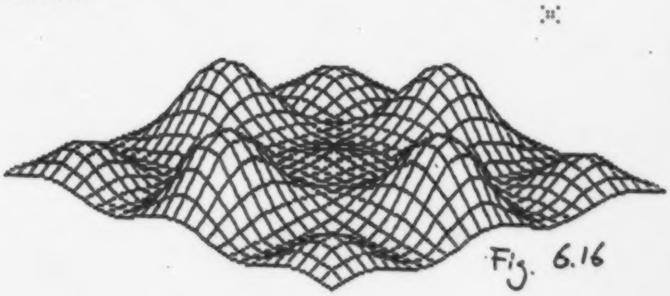
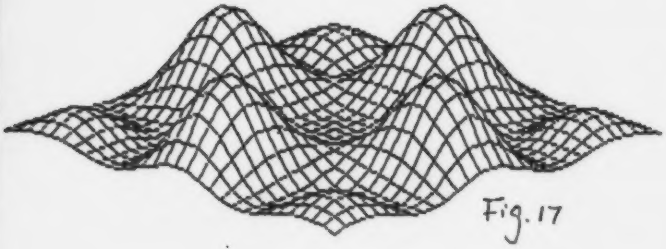


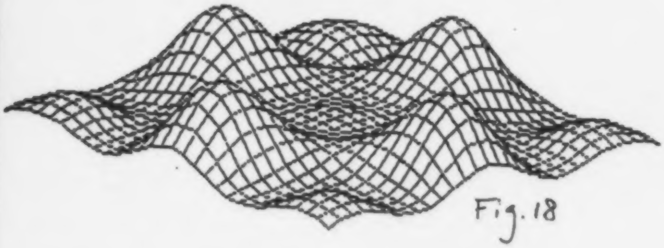
Fig. 6.16



TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6015.COL

Distance to Array:15mm

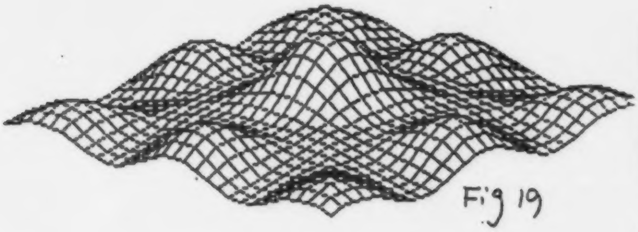
Fig. 17



TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6010.COL

Distance to Array:10mm

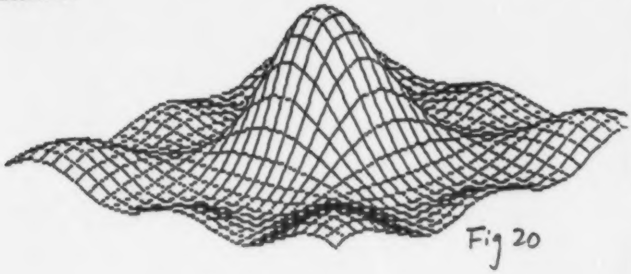
Fig. 18



TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6008.COL

Distance to Array:8mm

Fig 19



TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP6006.COL

Distance to Array:6mm

Fig 20

Fig. 6.1 – 6.20

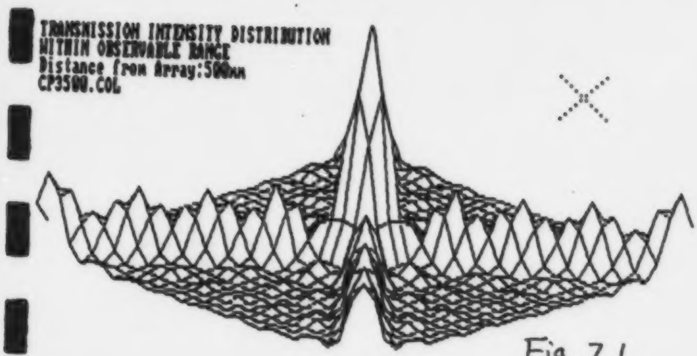


Fig. 7.1

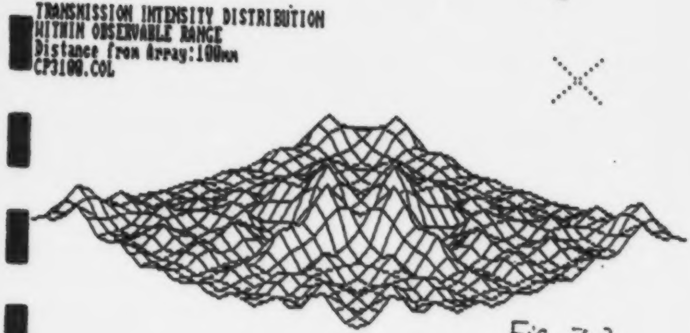


Fig. 7.2

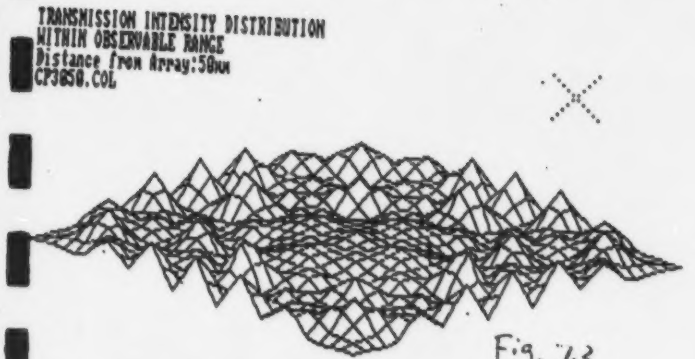


Fig. 7.3

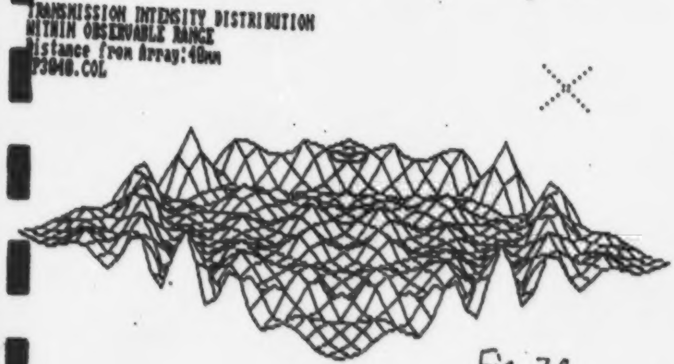


Fig. 7.4

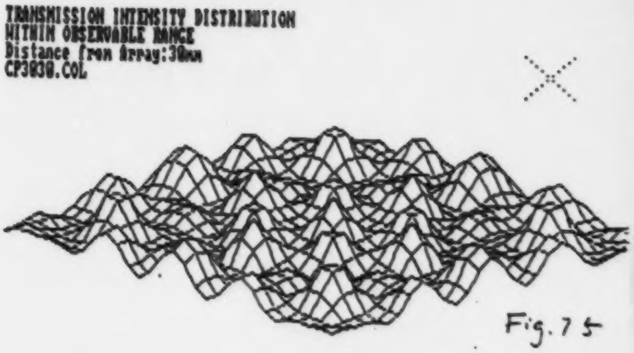


Fig. 7.5

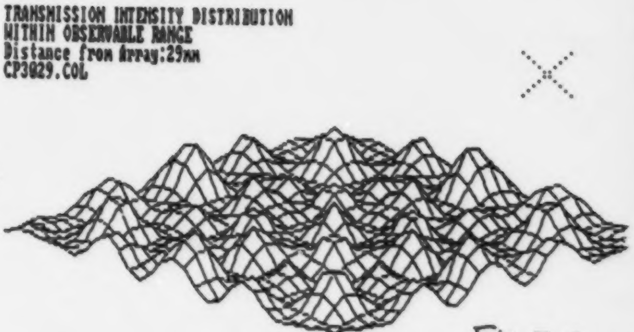


Fig. 7.6

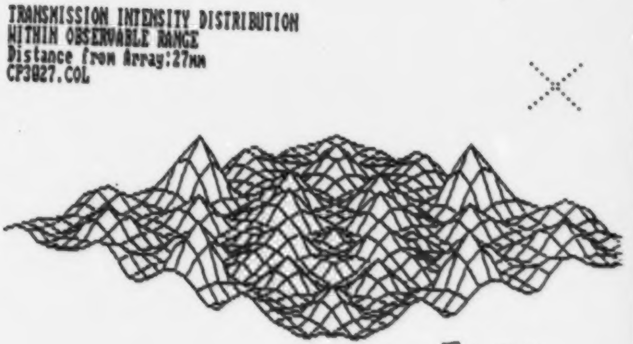


Fig. 7.7

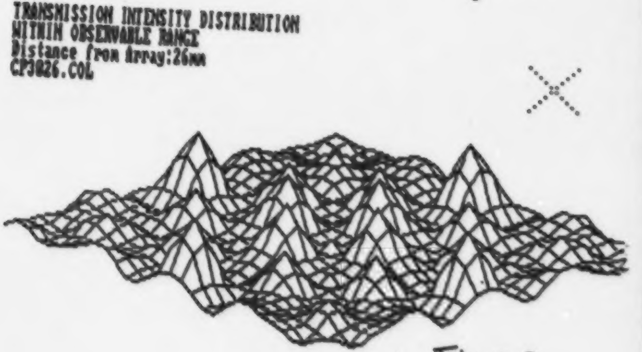


Fig. 7.8

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:25mm
CP3025.COL

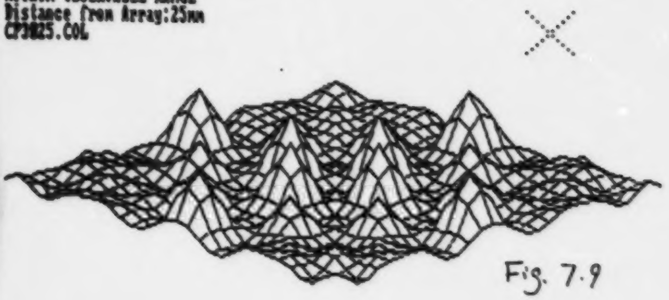


Fig. 7.9

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:24mm
CP3024.COL

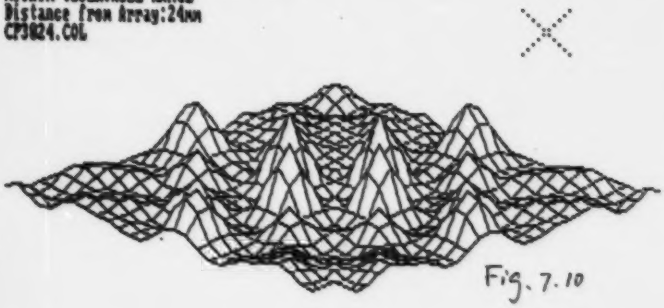


Fig. 7.10

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:23mm
CP3023.COL

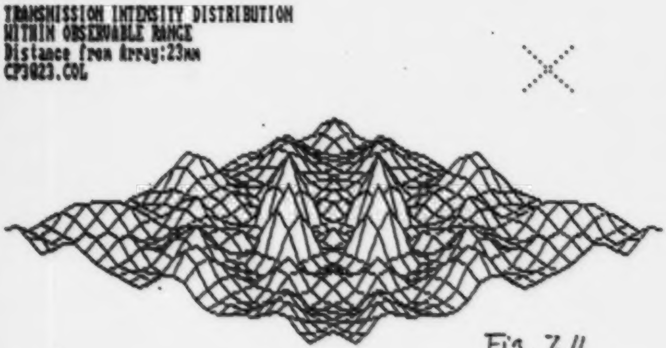


Fig. 7.11

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:22mm
CP3022.COL

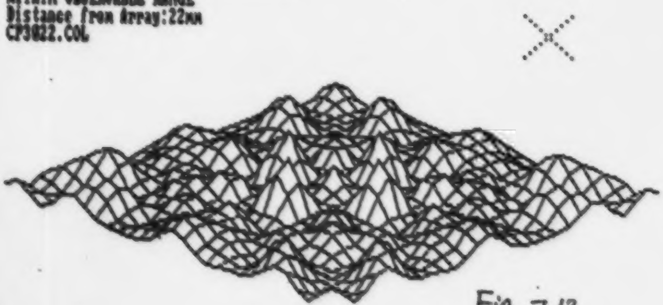


Fig. 7.12

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:21mm
CP3021.COL

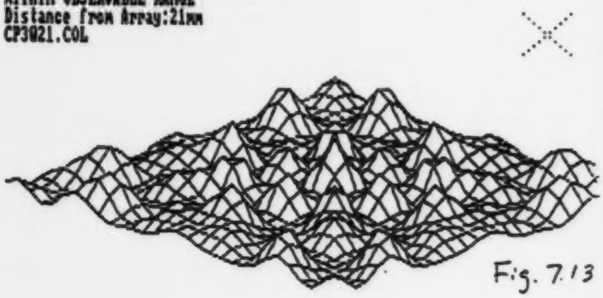


Fig. 7.13

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:20mm
CP3020.COL

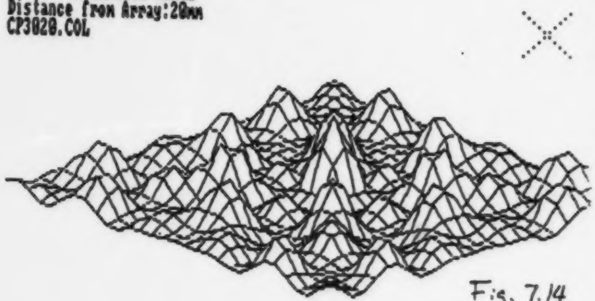


Fig. 7.14

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:19mm
CP3019.COL

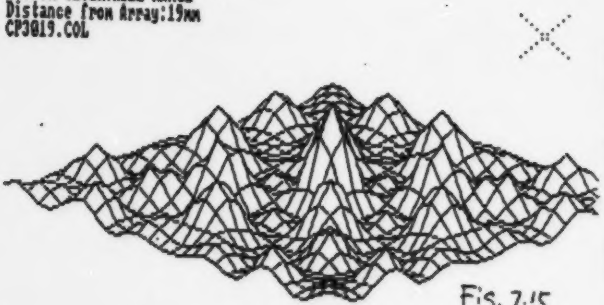


Fig. 7.15

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:18mm
CP3018.COL

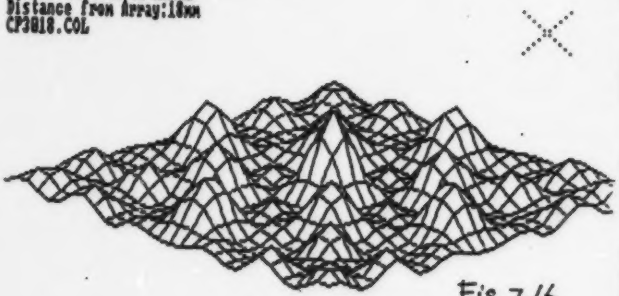


Fig. 7.16

TRANSMISSION INTENSITY DISTRIBUTION
ITHIN OBSERVABLE RANGE
istance from Array:15mm
P3015.COL

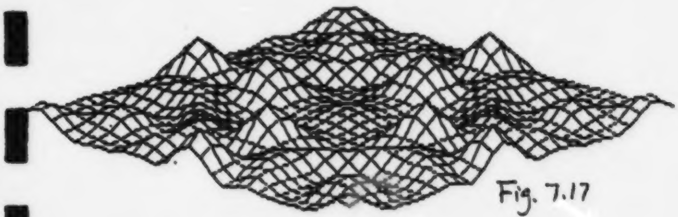


Fig. 7.17

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:10mm
CP3010.COL

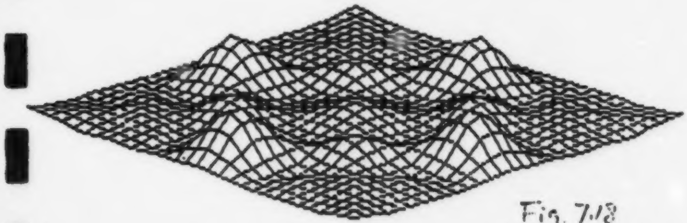
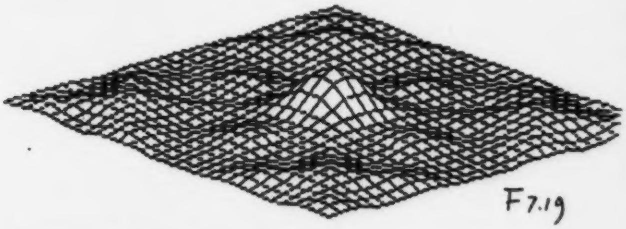


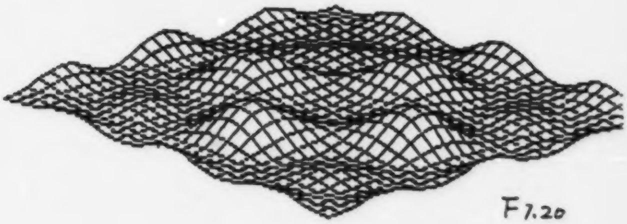
Fig. 7.18

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:8mm
CP3008.COL



F7.19

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
Distance from Array:6mm
CP3006.COL



F7.20

Fig. 7.1–7 20

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4030.COL

Distance to Array:30mm

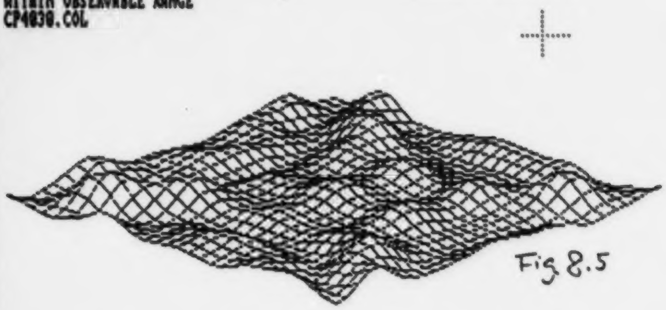


Fig. 8.5

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4029.COL

Distance to Array:29mm

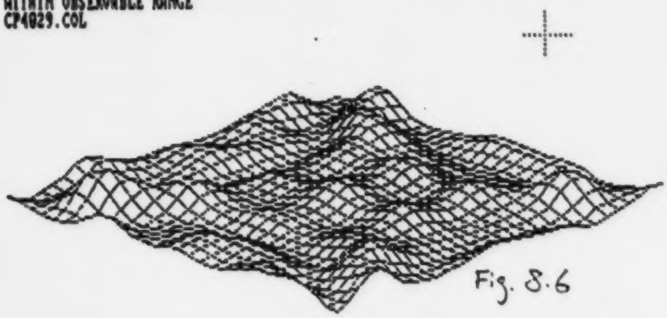


Fig. 8.6

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4027.COL

Distance to Array:27mm

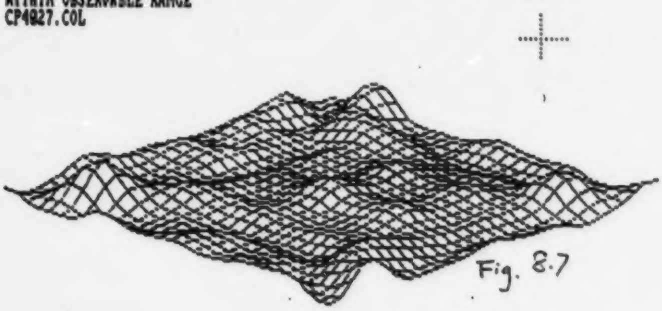


Fig. 8.7

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4026.COL

Distance to Array:26mm

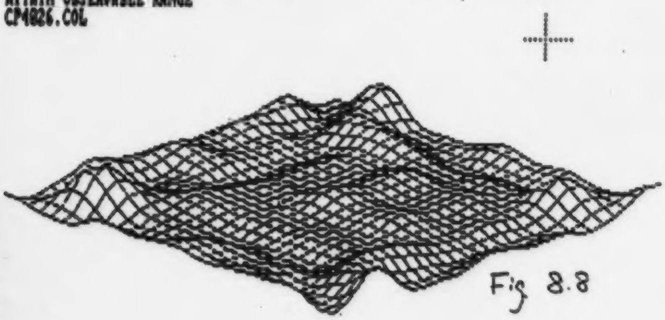


Fig 8.8

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4500.COL

Distance to Array:500mm

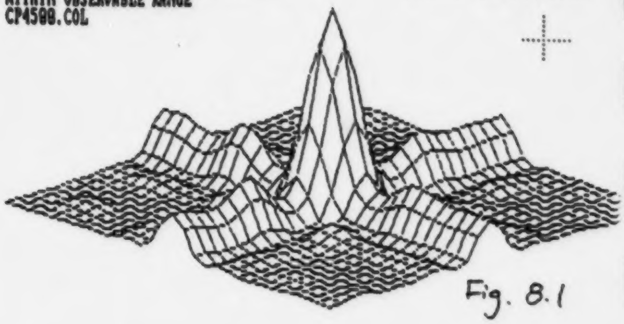


Fig. 8.1

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4100.COL

Distance to Array:100mm

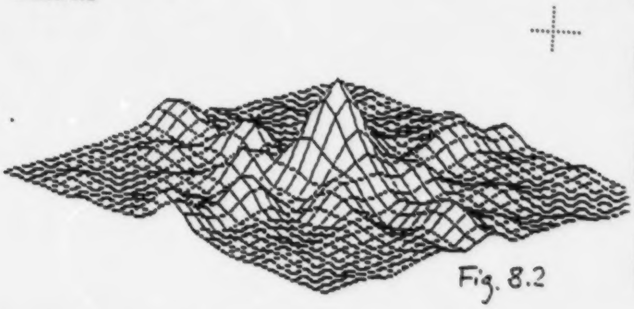


Fig. 8.2

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4050.COL

Distance to Array:50mm

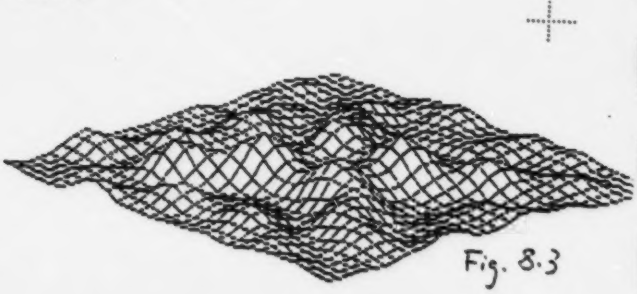


Fig. 8.3

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4040.COL

Distance to Array:40mm

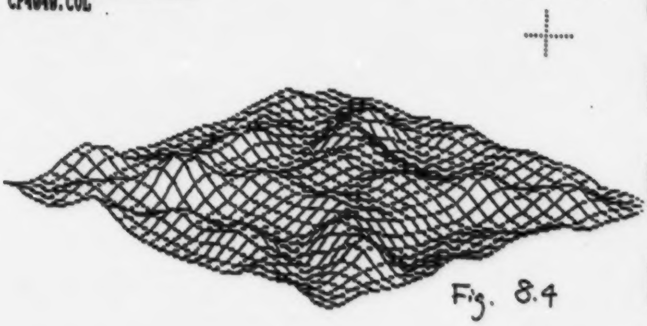


Fig. 8.4

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4021.COL

Distance to Array:21mm

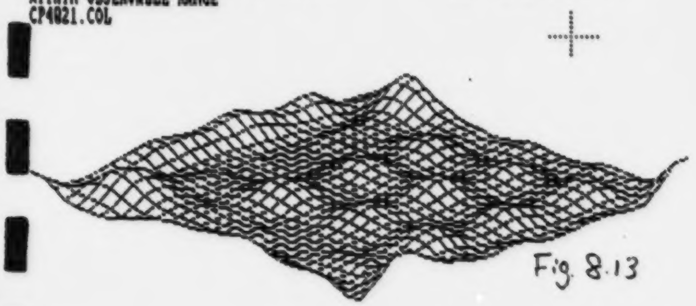


Fig. 8.13

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4020.COL

Distance to Array:20mm

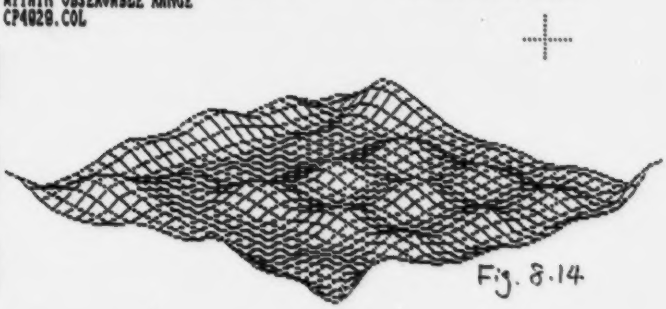


Fig. 8.14

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4019.COL

Distance to Array:19mm

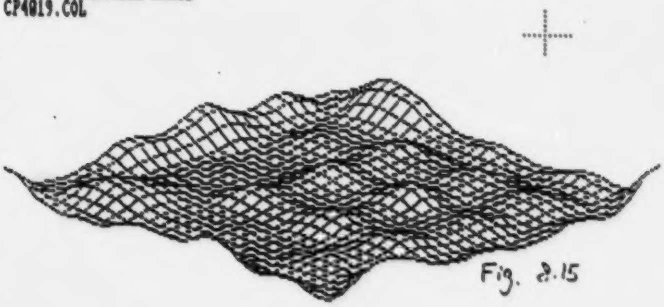


Fig. 8.15

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4018.COL

Distance to Array:18mm

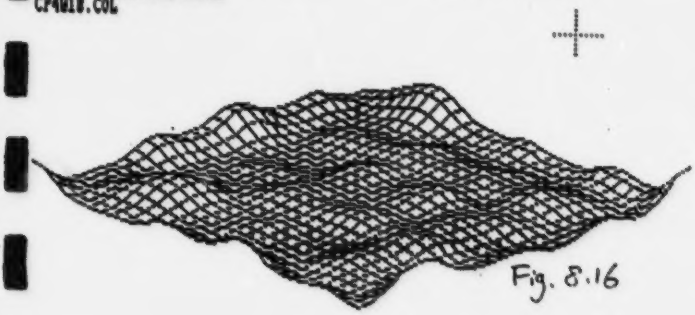


Fig. 8.16

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4025.COL

Distance to Array:25mm

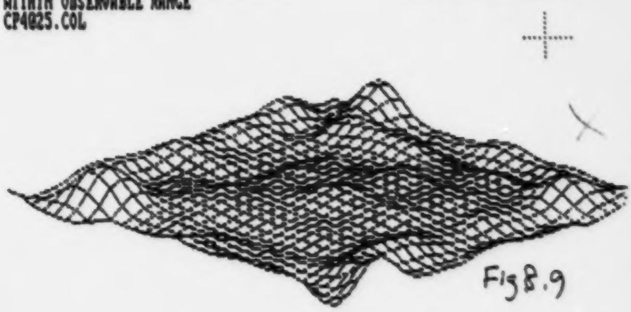


Fig8.9

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4024.COL

Distance to Array:24mm

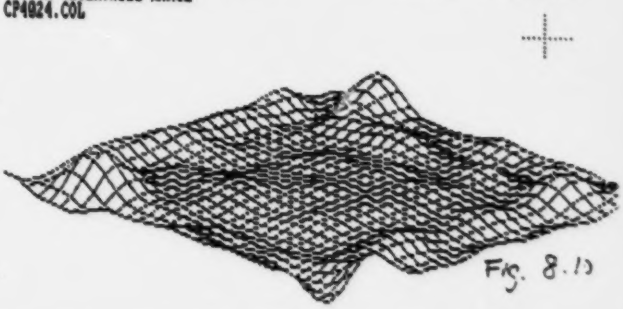


Fig. 8.10

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4023.COL

Distance to Array:23mm

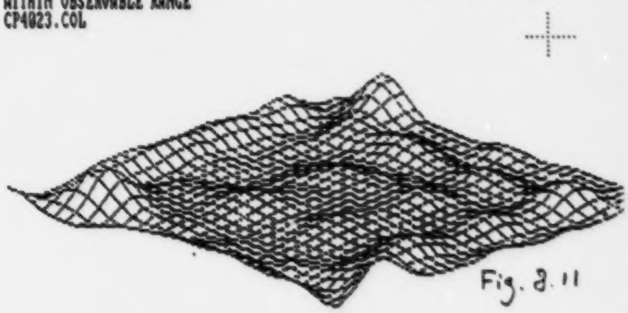


Fig. 8.11

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4022.COL

Distance to Array:22mm

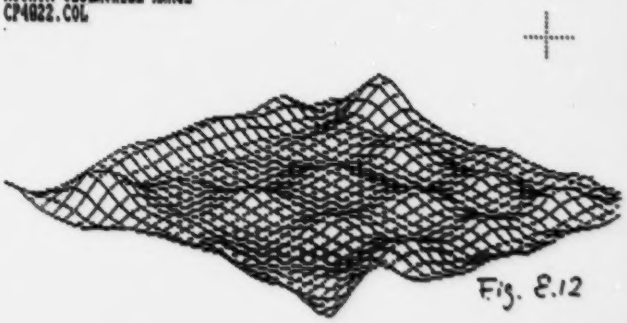


Fig. 8.12

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4008.COL

Distance to Array:8mm

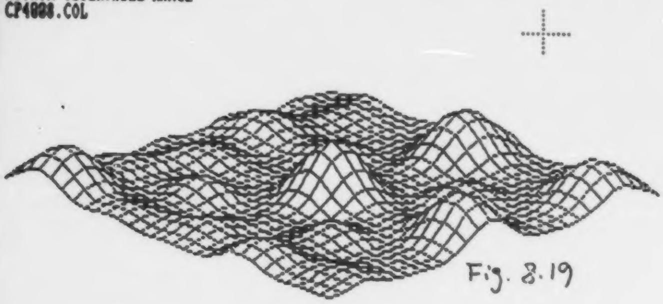


Fig. 8.19

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4015.COL

Distance to Array:15mm

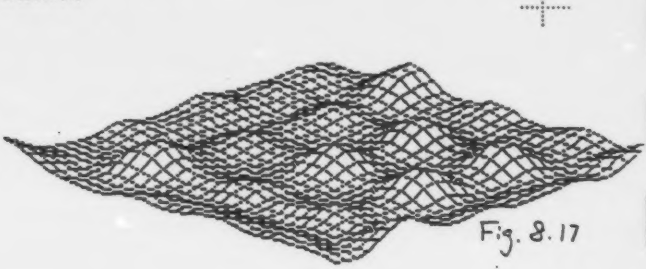


Fig. 8.17

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4006.COL

Distance to Array:6mm

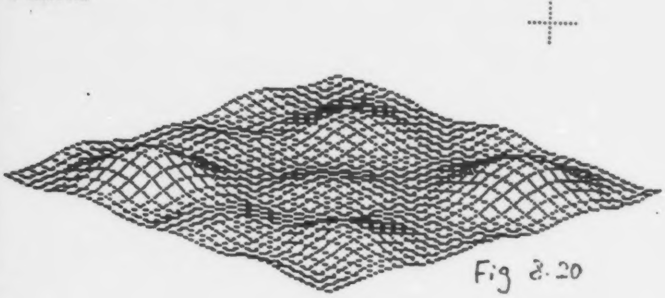


Fig 8.20

TRANSMISSION INTENSITY DISTRIBUTION
WITHIN OBSERVABLE RANGE
CP4010.COL

Distance to Array:10mm

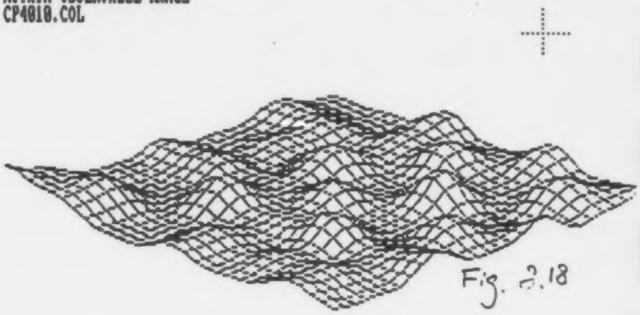
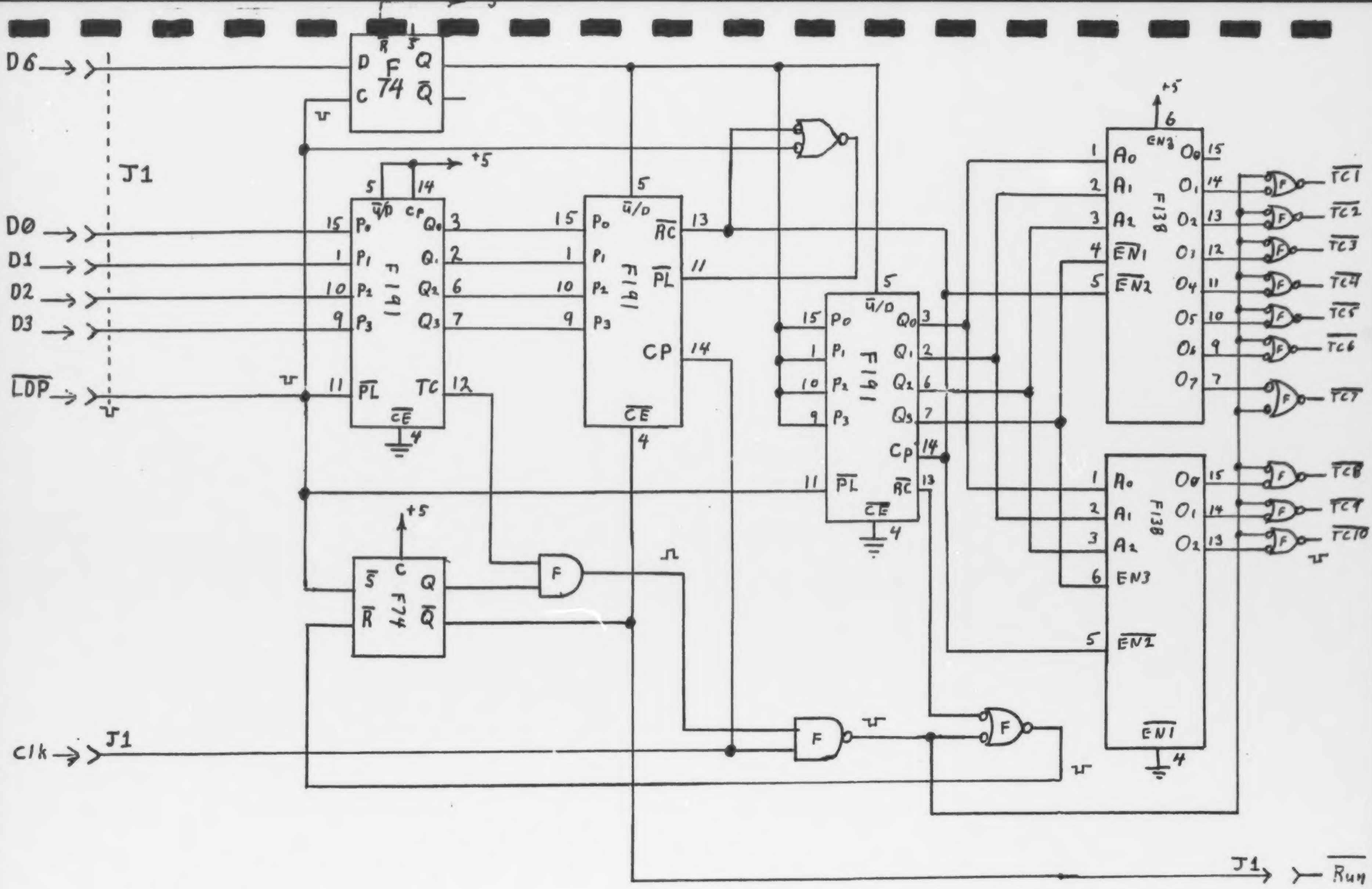


Fig. 8.18

Fig. 8.1 — 8.20

APPENDIX C

ELECTRONICS CIRCUIT DIAGRAM

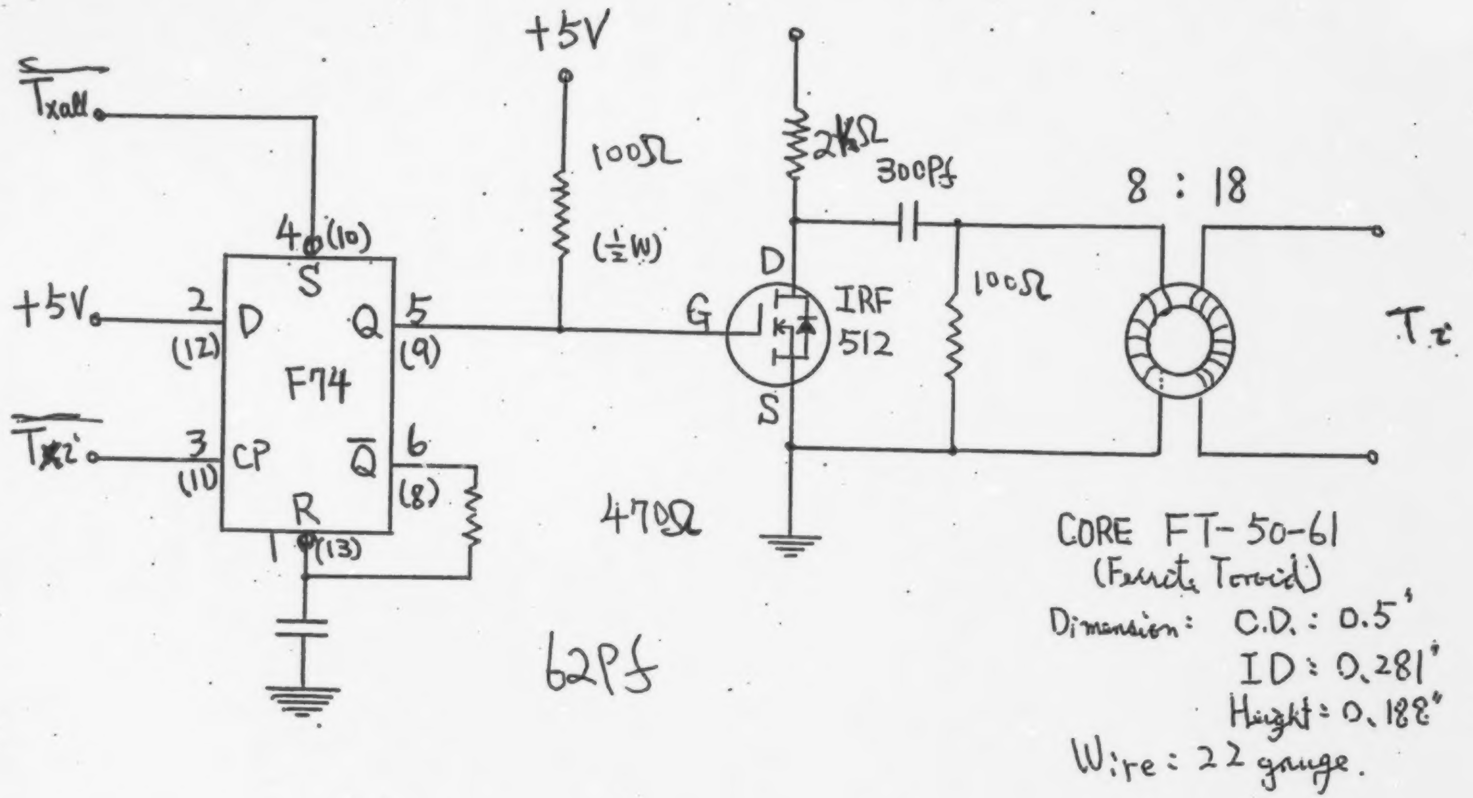

D6
D0
D1
D2
D3
LDP
Clk
J1
J1
J1
Run
F74
F191
F191
F191
F138
F138
F74
F
+5
TC1
TC2
TC3
TC4
TC5
TC6
TC7
TC8
TC9
TC10

HUGH W JONES & ASSOCIATES LTD

designed TEH
drawn kpn
date Aug 1986
revised

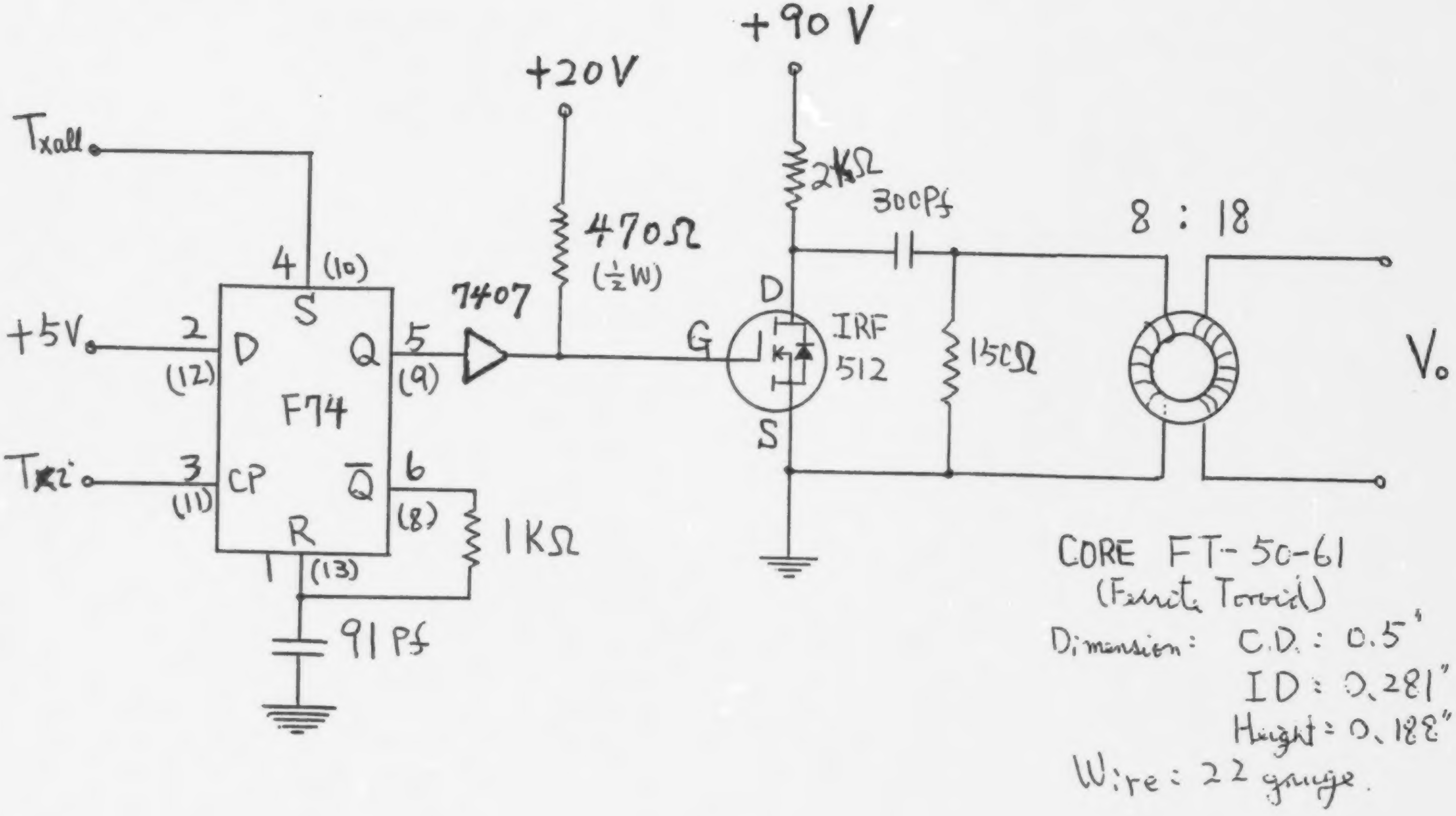
NDT CONTRACT
Phased Array Assenbly
Transmitting Signal
Timing Circuit

sheet 1 of 2


Txall
+5V
Txi
2
(12)
3
(11)
4
(10)
S
D
CP
R
Q
Q̄
F74
5
(9)
6
(8)
1
(13)
+5V
100Ω
(½W)
470Ω
62Pf
G
D
S
IRF
512
2KΩ
300Pf
100Ω
8 : 18
Tx
CORE FT-50-61
(Ferrite Toroid)
Dimension: O.D. : 0.5"
ID : 0.281"
Height : 0.188"
Wire : 22 gauge.

Original OUtput Driver Circuitry Design

HUGH W JONES & ASSOCIATES LTD	
designed HWK	NDT CONTRACT
drawn kpn	Phased Array Assembly Transmitting Signal Driver Circuit
date Aug 1986	
revised Sept. 26/86	sheet 1 of 1

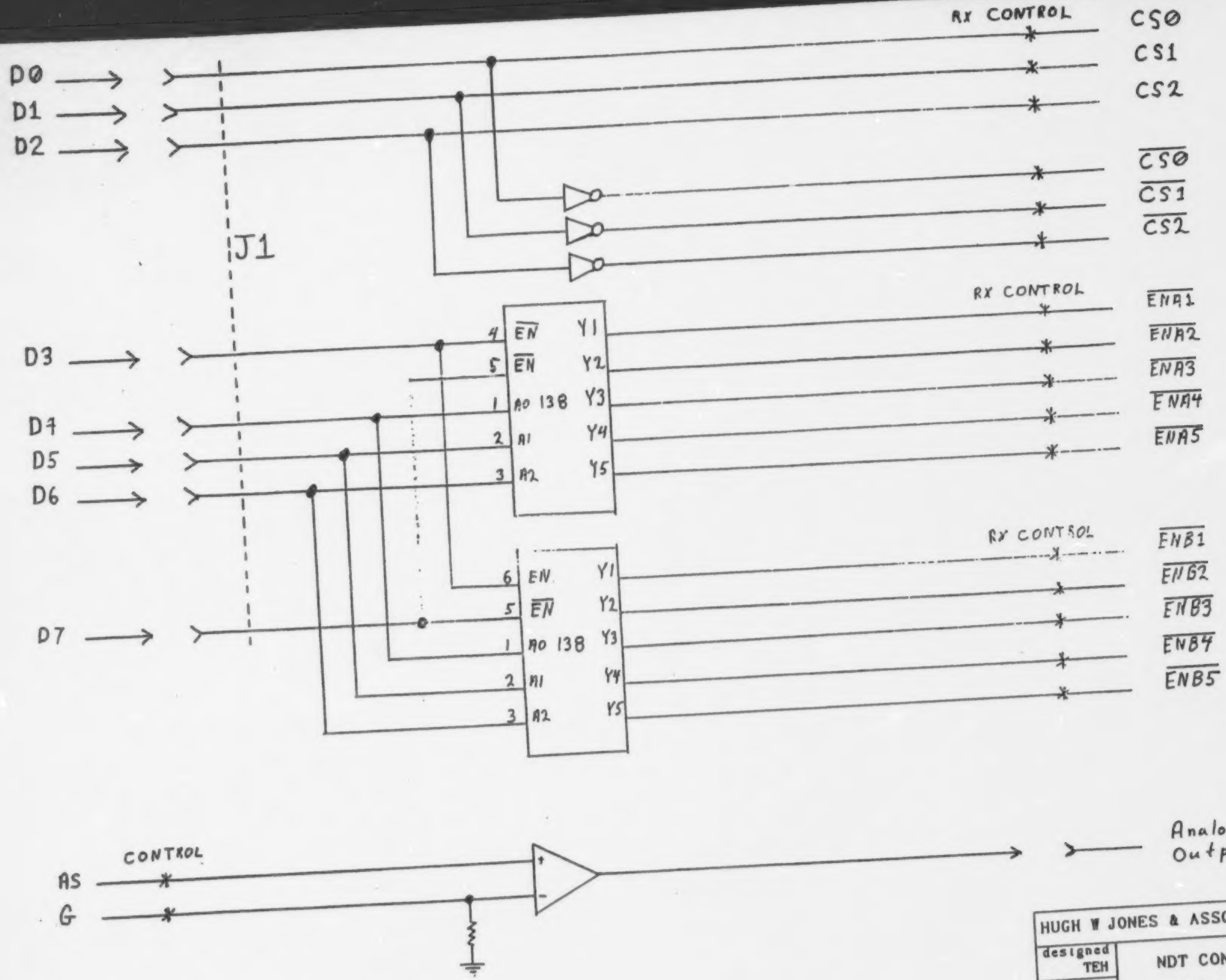


Txall
+5V
Tx2
4 (10)
2 (12)
3 (11)
S
D
CP
R
Q
Q̄
F74
5 (9)
6 (8)
1 (13)
7407
1KΩ
91 Pf
+20V
470Ω
(½W)
+90 V
2KΩ
300Pf
G
D
S
IRF
512
150Ω
8 : 18
V₀
CORE FT-50-61
(Ferrite Toroid)
Dimension: O.D.: 0.5"
ID: 0.281"
Height: 0.188"
Wire: 22 gauge.

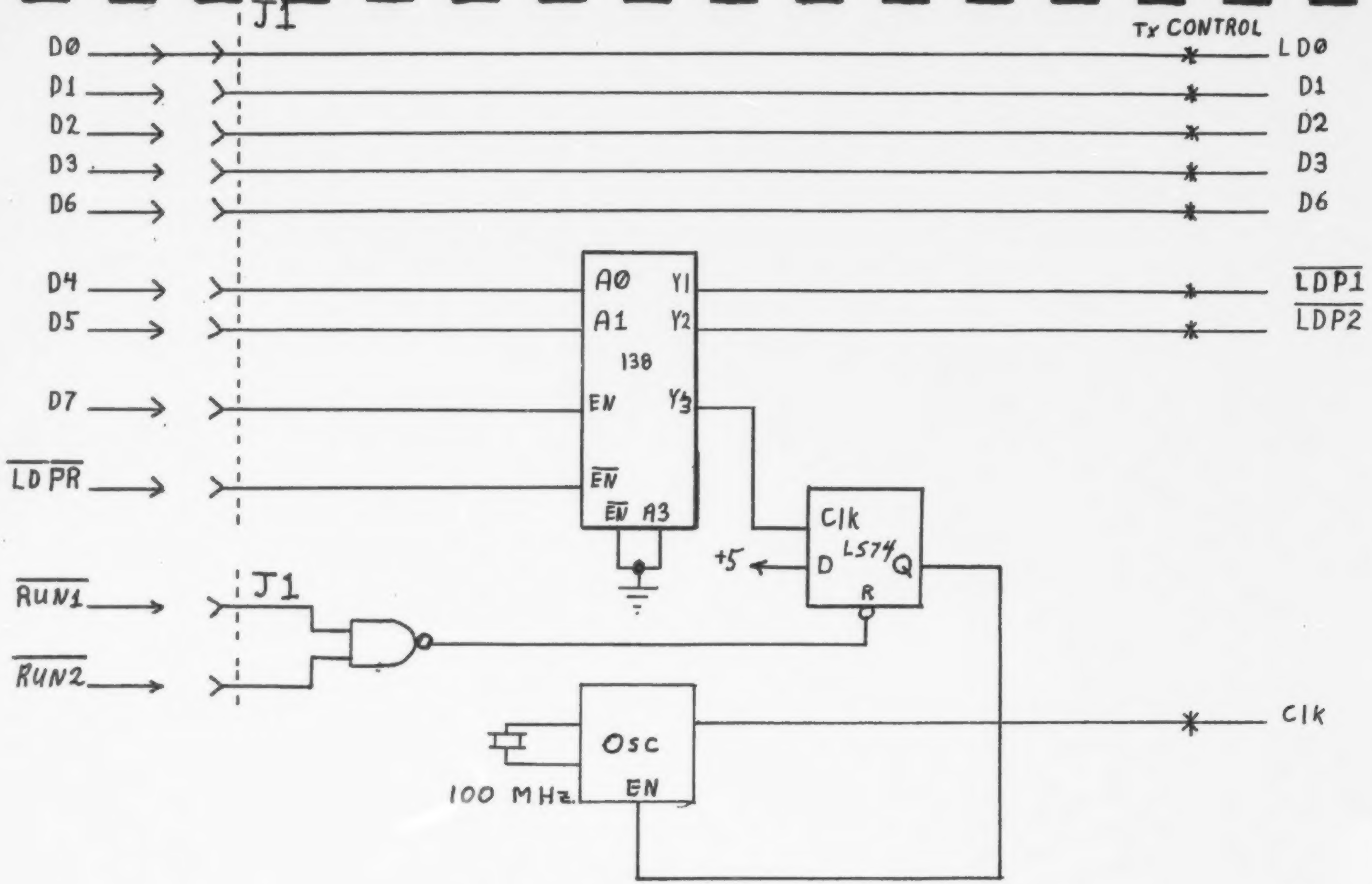
Revised Output Driver Circuitry Design

HUGH W JONES & ASSOCIATES LTD

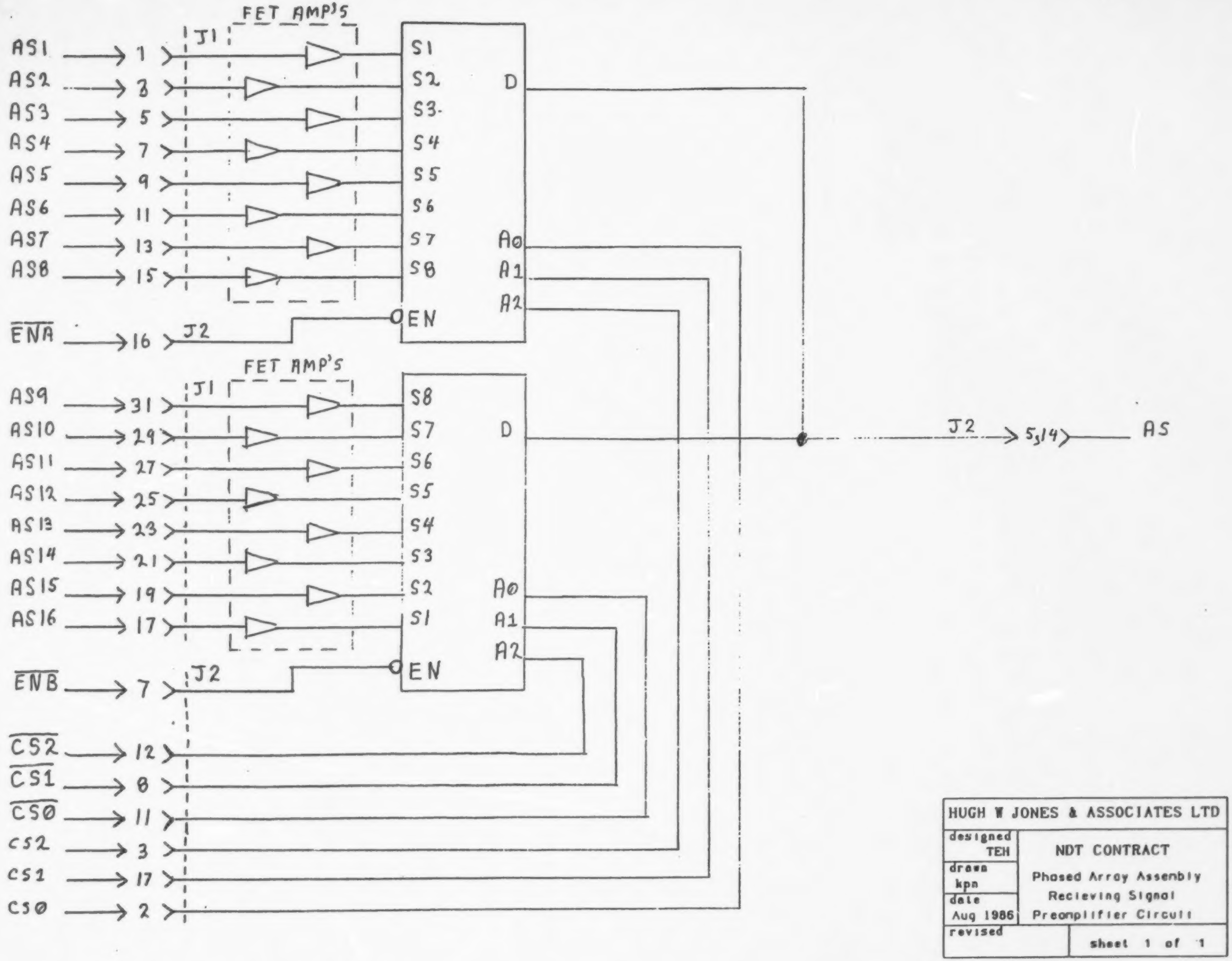
designed HWK	NDT CONTRACT
drawn kpn	Phased Array Assembly Transmitting Signal Driver Circuit
date Aug 1986	
revised	sheet 1 of 1


D0
D1
D2
D3
D4
D5
D6
D7
J1
RX CONTROL
CS0
CS1
CS2
$\overline{CS0}$
$\overline{CS1}$
$\overline{CS2}$
RX CONTROL
4 $\overline{EN}$
5 $\overline{EN}$
1 A0 138
2 A1
3 A2
Y1
Y2
Y3
Y4
Y5
$\overline{ENA1}$
$\overline{ENA2}$
$\overline{ENA3}$
$\overline{ENA4}$
$\overline{ENA5}$
RX CONTROL
6 EN
5 $\overline{EN}$
1 A0 138
2 A1
3 A2
Y1
Y2
Y3
Y4
Y5
$\overline{ENB1}$
$\overline{ENB2}$
$\overline{ENB3}$
$\overline{ENB4}$
$\overline{ENB5}$
CONTROL
AS
G
+
-
Analog Output

HUGH W JONES & ASSOCIATES LTD		
designed TEH	NDT CONTRACT	
drawn kpn	Phased Array Assembly Control Bus	
date Aug 1986	Interface Circuit	
revised		sheet 1 of 2


J1
Tx CONTROL
D0
D1
D2
D3
D6
D4
D5
D7
$\overline{LDPR}$
$\overline{RUN1}$
$\overline{RUN2}$
LD0
D1
D2
D3
D6
$\overline{LDP1}$
$\overline{LDP2}$
A0
A1
Y1
Y2
Y3
138
EN
$\overline{EN}$
$\overline{EN}$
A3
Clk
D
LS74
Q
R
+5
Osc
EN
100 MHz
Clk

HUGH W JONES & ASSOCIATES LTD	
designed TEH	NDT CONTRACT
drawn kpn	Phased Array Assenbly Control Bus Interface Circuit
date Aug 1986	
revised	sheet 2 of 2


FET AMP'S
J1
AS1 → 1
AS2 → 3
AS3 → 5
AS4 → 7
AS5 → 9
AS6 → 11
AS7 → 13
AS8 → 15
$\overline{ENA}$ → 16
J2
S1
S2
S3
S4
S5
S6
S7
S8
EN
D
A0
A1
A2
FET AMP'S
J1
AS9 → 31
AS10 → 29
AS11 → 27
AS12 → 25
AS13 → 23
AS14 → 21
AS15 → 19
AS16 → 17
$\overline{ENB}$ → 7
J2
S8
S7
S6
S5
S4
S3
S2
S1
EN
D
A0
A1
A2
$\overline{CS2}$ → 12
$\overline{CS1}$ → 8
$\overline{CS0}$ → 11
CS2 → 3
CS1 → 17
CS0 → 2
J2 → 5,14 — AS

HUGH W JONES & ASSOCIATES LTD

designed TEH

drawn kpn

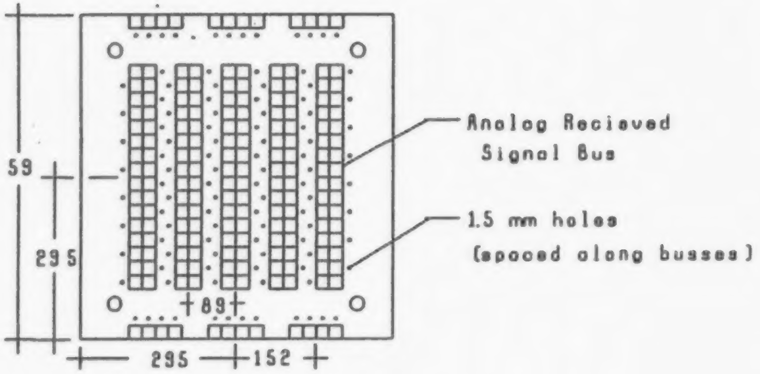
date Aug 1986

revised

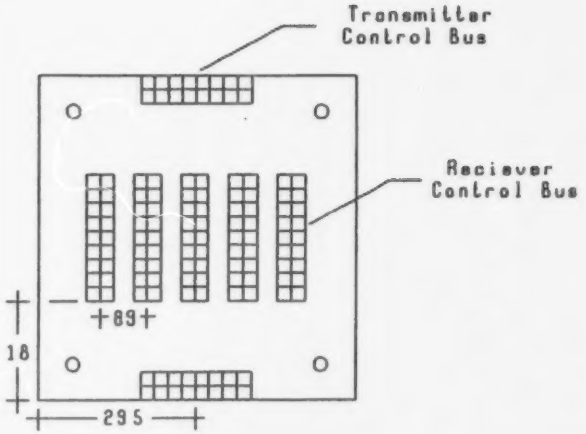
NDT CONTRACT

Phased Array Assenbly
Recieving Signal
Preamplifier Circuit

sheet 1 of 1


59
29 5
295
152
89
Analog Recieved
Signal Bus
1.5 mm holes
(spaced along busses)

SIGNAL BUS CARD

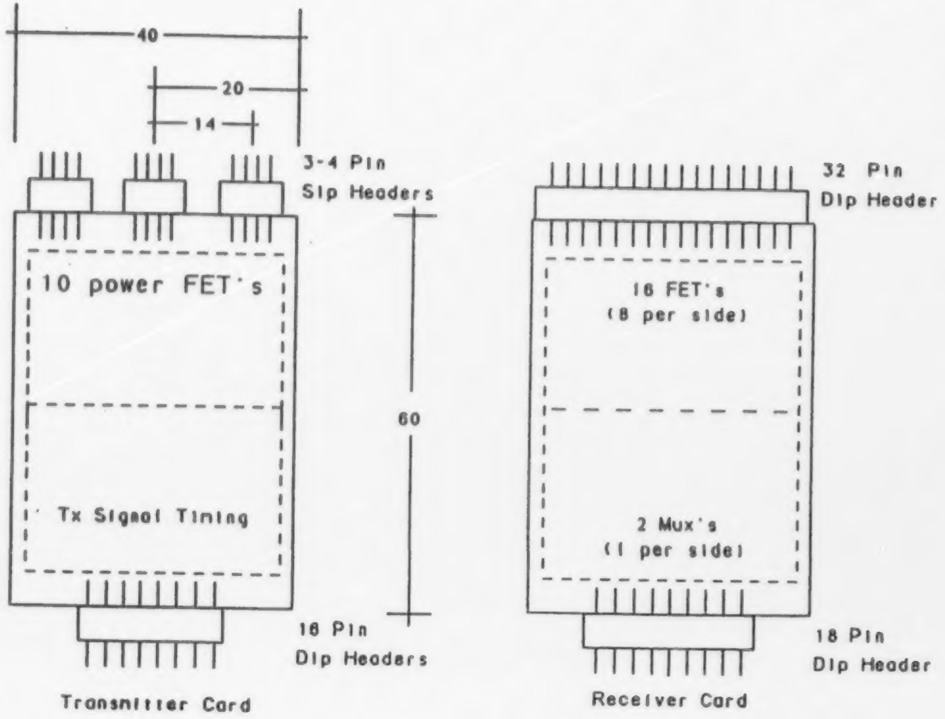

Transmitter
Control Bus
Reciever
Control Bus
18
89
29 5

CONTROL BUS CARD

*

measurment in millimeters

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpa	Phased Array Assembly
date Aug 1986	Bus Cards
revised	sheet 6 of 7


40
20
14
3-4 Pin
Sip Headers
10 power FET's
Tx Signal Timing
60
18 Pin
Dip Headers
Transmitter Card
32 Pin
Dip Header
16 FET's
(8 per side)
2 Mux's
(1 per side)
18 Pin
Dip Header
Receiver Card

measurment in millimeters

headers centered on boards unless indicated

HUGH W JONES & ASSOCIATES LTD	
designed HWK-TEH	NDT CONTRACT
drawn kpn	Phased Array Assembly
date Aug 1986	Transmit & Recieve Cards
revised	sheet 7 of 7

ANALOG SYSTEM - Phased array electronics
Proposed Card connector pinouts:

Pin	Side A	Side B	Pin
1	AS1	G	32
	G	AS9	
	AS2	G	
4	G	AS10	
	AS3	G	28
	G	AS11	
	AS4	G	
8	G	AS12	
	AS5	G	24
	G	AS13	
	AS6	G	
12	G	AS14	
	AS7	G	20
	G	AS15	
	AS8	G	
16	G	AS16	17

Signal
Connector

(J1)

1	+V	-V	18
	CS0	CS1	
	CS2	$\overline{ENA}$	16
4	GND	G	
	AS	AS	
	G	GND	
	$\overline{ENB}$	$\overline{CS2}$	12
8	$\overline{CS1}$	$\overline{CS0}$	
9	-V	+V	

Control
Connector

(J2)

(G ≡ signal ground)

ANALOG SYSTEM - Phased array electronics
Proposed Control bus:

bus line	+12V DC	+V	-V	-12V DC	bus line
bus line	channel select	A0	A1	channel select	bus line
bus line	channel select	A2	$\overline{ENA}$	bank A enable	individual contact
bus line	power ground	Gnd	G	signal ground	bus line
bus line	analog signal	AS	AS	analog signal	bus line
bus line	signal ground	G	Gnd	power ground	bus line
individual contact	bank B enable	$\overline{ENB}$	$\overline{A2}$	channel select	bus line
bus line	channel select	$\overline{A1}$	$\overline{A0}$	channel select	bus line
bus line	-12V DC	-V	+V	+12V DC	bus line

ANALOG SYSTEM - Phased array electronics
Proposed Signal bus:

patch contact	signal	AS1	G	ground	bus line
bus line	ground	G	AS9	signal	patch contact
patch contact	signal	AS2	G	ground	bus line
bus line	ground	G	AS10	signal	patch contact
patch contact	signal	AS3	G	ground	bus line
bus line	ground	G	AS11	signal	patch contact
patch contact	signal	AS4	G	ground	bus line
bus line	ground	G	AS12	signal	patch contact
patch contact	signal	AS5	G	ground	bus line
bus line	ground	G	AS13	signal	patch contact
patch contact	signal	AS6	G	ground	bus line
bus line	ground	G	AS14	signal	patch contact
patch contact	signal	AS7	G	ground	bus line
bus line	ground	G	AS15	signal	patch contact
patch contact	signal	AS8	G	ground	bus line
	ground	G	AS16	signal	patch contact

APPENDIX D

SOFTWARE LISTINGS

```
                                PAGE
0000                            CSEG    SEGMENT
                                        ASSUME  CS:CSEG
0000                            START:
                                ;               V E C T O R   T A B L E :
0000  E9 0183 R                                 JMP     INIT
0003  E9 01DA R                                 JMP     CAMACOM
0006  E9 01E9 R                                 JMP     Get1block
                                ;
                                ;               P R O G R A M   E Q U A T E S :
                                ;
= 000A                          CRATE_ADDR      EQU     10      ;GPIB address of camac crate
```

```
                                  PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                  ;                                                                               :
                                  ;               P R O C E D U R E   C O M M A N D 8 2 9 2                       :
                                  ;                                                                               :
                                  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                  ;
                                  ;INPUTS:        command byte in AH
                                  ;OUTPUTS:       none
                                  ;CALLS:         none
                                  ;DESTROYS:      AL, DX & Flags
                                  ;
0009                              CMD8292S        PROC    NEAR
                                                  WAIT_COM                  ;wait until 8292 ready for next command
0009  BA 0309                +                    MOV     DX,INTST
000C  EC                     + ??0000:            IN      AL,DX             ;get TCI status bit
000D  24 02                  +                    AND     AL,IBFET
000F  75 FB                  +                    JNZ     ??0000            ;wait until false
                                                  OUTB    TCILC,0           ;Clear TCI Latch
0011  B0 00                  +                    MOV     AL,0
0013  BA 030A                +                    MOV     DX,TCILC
0016  EE                     +                    OUT     DX,AL
0017  BA 0309                                     MOV     DX,CMD92
001A  8A C4                                       MOV     AL,AH             ;get command byte
001C  EE                                          OUT     DX,AL             ;& output it to 8292
                                                  WAITT
001D  BA 030C                +                    MOV     DX,PRTIDA
0020  EC                     + ??0001:            IN      AL,DX             ;get TCI status bit
0021  24 01                  +                    AND     AL,TCIM
0023  74 FB                  +                    JZ      ??0001            ;wait until false
0025  C3                                          RET
0026                              CMD8292S        ENDP

0026                              CMD8292         PROC    NEAR
                                                  OUTB    TCILC,0           ;Clear TCI Latch
0026  B0 00                  +                    MOV     AL,0
0028  BA 030A                +                    MOV     DX,TCILC
002B  EE                     +                    OUT     DX,AL
002C  BA 0309                                     MOV     DX,CMD92
002F  8A C4                                       MOV     AL,AH             ;get command byte
0031  EE                                          OUT     DX,AL             ;& output it to 8292
0032  51                                          PUSH    CX
0033  B9 0078                                     MOV     CX,078H
0036  E2 FE                                       LOOP    $
0038  59                                          POP     CX
0039  C3                                          RET
003A                              CMD8292         ENDP
```

```
                                  PAGE: -------------------------------------------------:
                                  :                                                       :
                                  :            P R O C E D U R E   C A M A C   C O M M A N D  :
                                  :                                                       :
                                  :-------------------------------------------------------:
                                  :
                                  ;INPUTS:        ES:SI points to command string
                                  ;OUTPUTS:       none
                                  ;CALLS:         COM8292
                                  ;DESTROYS:      AX, DX & Flags
                                  :
003A                              CAMACMND        PROC    NEAR
003A  51                                          PUSH    CX
003B  56                                          PUSH    SI
                                                  LISTEN_FST CRATE_ADDR   ;tell crate to listen
003C  BA 0301                   +                 MOV     DX,INT1
003F                            + ??0002:
003F  EC                        +                 IN      AL,DX           ;get int 1 status
0040  24 02                     +                 AND     AL,BOM          ;check for byte out
0042  74 FB                     +                 JZ      ??0002          ;if not, try again
0044  B0 0A                     +                 MOV     AL,CRATE_ADDR
0046  04 20                     +                 ADD     AL,020H
0048  BA 0300                   +                 MOV     DX,DOUT
004B  EE                        +                 OUT     DX,AL           ;& output to 8291
004C  26: 8A 0C                                   MOV     CL,ES:[SI]      ; & get string size
004F  80 F9 00                                    CMP     CL,0
0052  74 4C                                       JE      EndIfThen       ;send no data if nul string
0054                              BegIfThen:      OUT_MTA                 ;send my talk address
0054  BA 0301                   +                 MOV     DX,INT1
0057                            + ??0003:
0057  EC                        +                 IN      AL,DX           ;get int 1 status
0058  24 02                     +                 AND     AL,BOM          ;check for byte out
005A  74 FB                     +                 JZ      ??0003          ;if not, try again
005C  B0 41                     +                 MOV     AL,MTA
005E  BA 0300                   +                 MOV     DX,DOUT
0061  EE                        +                 OUT     DX,AL
0062  BA 0301                   +                 MOV     DX,INT1
0065                            + ??0004:
0065  EC                        +                 IN      AL,DX           ;get int 1 status
0066  24 02                     +                 AND     AL,BOM          ;check for byte out
0068  74 FB                     +                 JZ      ??0004          ;if not, try again
                                                  OUT_GTSB                ;go to stand by
006A  B4 F6                     +                 MOV     AH,GTSB
006C  E8 0026 R                 +                 CALL    CMD8292         ;GoToStandBy
006F  46                          OUTDATA:        INC     SI
0070  26: 8A 04                                   MOV     AL,ES:[SI]      ;send the data
                                                  WRITE_AL
0073  BA 0300                   +                 MOV     DX,DOUT
0076  EE                        +                 OUT     DX,AL           ;& output to 8291
0077  BA 0301                   +                 MOV     DX,INT1
007A                            + ??0005:
007A  EC                        +                 IN      AL,DX           ;get int 1 status
007B  24 02                     +                 AND     AL,BOM          ;check for byte out
007D  74 FB                     +                 JZ      ??0005          ;if not, try again
007F  FE C9                                       DEC     CL
```

```
0081  75 EC                               JNZ     OUTDATA
                                          OUT_TCSY              ;take control syncroniously
0083  B4 FD             +                 MOV     AH,TCSY
0085  E8 0026 R         +                 CALL    CMD8292
0088  B0 80             +                 MOV     AL,AXRA
008A  BA 0305           +                 MOV     DX,AUXMD
008D  EE                +                 OUT     DX,AL
008E  B0 03             +                 MOV     AL,FNHSK
0090  BA 0305           +                 MOV     DX,AUXMD
0093  EE                +                 OUT     DX,AL
0094  B0 80             +                 MOV     AL,TON
0096  BA 0304           +                 MOV     DX,ADRMD
0099  EE                +                 OUT     DX,AL
009A  B0 00             +                 MOV     AL,PON
009C  BA 0305           +                 MOV     DX,AUXMD
009F  EE                +                 OUT     DX,AL
00A0                      EndIfThen:      TALK    CRATE_ADDR
00A0  BA 0301           +                 MOV     DX,INT1
00A3                    + ??0006:
00A3  EC                +                 IN      AL,DX         ;get int 1 status
00A4  24 02             +                 AND     AL,BOM        ;check for byte out
00A6  74 FB             +                 JZ      ??0006        ;if not, try again
00A8  B0 0A             +                 MOV     AL,CRATE_ADDR ;get address byte
00AA  04 40             +                 ADD     AL,040H
00AC  BA 0300           +                 MOV     DX,DOUT
00AF  EE                +                 OUT     DX,AL         ;& output it to 8292
00B0  5E                                  POP     SI
00B1  59                                  POP     CX
00B2  C3                                  RET
00B3                      CAMACMND        ENDP

00B3                      CAMACMNDS       PROC    NEAR
00B3  51                                  PUSH    CX
00B4  56                                  PUSH    SI
                                          ;LISTEN_FST CRATE_ADDR ;tell crate to listen
00B5  B0 2A                               MOV     AL,CRATE_ADDR+020H
00B7  BA 0300                             MOV     DX,DOUT
00BA  EE                                  OUT     DX,AL
00BB  26: 8A 0C                           MOV     CL,ES:[SI]      ; & get string size
00BE  32 ED                               XOR     CH,CH
00C0  E3 32                               JCXZ    EndIfThenS      ;send no data if nul string
00C2                      BegIfThenS:     ;OUT_MTA                ;send my talk address
                                          OUTB    DOUT,MTA
00C2  B0 41             +                 MOV     AL,MTA
00C4  BA 0300           +                 MOV     DX,DOUT
00C7  EE                +                 OUT     DX,AL
                                          OUT_GTSB                ;go to stand by
00C8  B4 F6             +                 MOV     AH,GTSB
00CA  E8 0026 R         +                 CALL    CMD8292         ;GoToStandBy
00CD  BA 0300                             MOV     DX,DOUT
00D0  46                  OUTDATAS:       INC     SI
00D1  26: 8A 04                           MOV     AL,ES:[SI]      ;send the data
00D4  EE                                  OUT     DX,AL
00D5  E2 F9                               LOOP    OUTDATAS
                                          OUT_TCSY                ;take control syncroniously
```

```
00D7  B4 FD                 +                   MOV    AH,TCSY
00D9  E8 0026 R             +                   CALL   CMD8292
00DC  B0 80                 +                   MOV    AL,AXRA
00DE  BA 0305               +                   MOV    DX,AUXMD
00E1  EE                    +                   OUT    DX,AL
00E2  B0 03                 +                   MOV    AL,FNHSK
00E4  BA 0305               +                   MOV    DX,AUXMD
00E7  EE                    +                   OUT    DX,AL
00E8  B0 80                 +                   MOV    AL,TON
00EA  BA 0304               +                   MOV    DX,ADRMD
00ED  EE                    +                   OUT    DX,AL
00EE  B0 00                 +                   MOV    AL,PON
00F0  BA 0305               +                   MOV    DX,AUXMD
00F3  EE                    +                   OUT    DX,AL
00F4                          EndIfThenS:       ;TALK  CRATE_ADDR
00F4  B0 4A                                     MOV    AL,CRATE_ADDR+040H
00F6  BA 0300                                   MOV    DX,DOUT
00F9  EE                                        OUT    DX,AL
00FA  5E                                        POP    SI
00FB  59                                        POP    CX
00FC  C3                                        RET
00FD                          CAMACMNDS         ENDP
```

```
                                    PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;                                                                                   :
                                    ;               H O L D   W H I L E   T R 8 8 3 7   D I G I T I Z I N G             :
                                    ;                                                                                   :
                                    ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;
                                    ;INPUTS:        none
                                    ;OUTPUTS:       none
                                    ;CALLS:         CMD8292, CAMACMND
                                    ;DESTROYS:      AX, ???
                                    ;
00FD                                HLDFOR8837      PROC    NEAR
00FD  26: C6 04 03                                  MOV     BYTE PTR ES:[SI],3        ;command string size
0101  26: C6 44 01 1B                               MOV     BYTE PTR ES:[SI+1],TstLAM ;  function=test lam via Q flag
0106  26: C6 44 02 00                               MOV     BYTE PTR ES:[SI+2],0              ;channel #0
010B  26: C6 44 03 0B                               MOV     BYTE PTR ES:[SI+3],Slot8837     ;camac slot # for 8837
0110  E8 003A R                                     CALL    CAMACMND

0113  26: C6 04 01                                  MOV     BYTE PTR ES:[SI],1        ;command string size
0117  26: C6 44 01 61                               MOV     BYTE PTR ES:[SI+1],ByteRet + Ret2Byte
011C  E8 003A R                                     CALL    CAMACMND

                                    ;               MOV     BYTE PTR ES:[SI],0        ;command string size
011F                                HOLDHERE:
011F  E8 003A R                                     CALL    CAMACMND                  ;execute camac cycle
                                                    OUT_MLA                           ; & read two bytes
0122  BA 0301                   +                   MOV     DX,INT1
0125                            + ??0007:
0125  EC                        +                   IN      AL,DX           ;get int 1 status
0126  24 02                     +                   AND     AL,BOM          ;check for byte out
0128  74 FB                     +                   JZ      ??0007          ;if not, try again
012A  B0 21                     +                   MOV     AL,MLA
012C  BA 0300                   +                   MOV     DX,DOUT
012F  EE                        +                   OUT     DX,AL
0130  B0 82                     +                   MOV     AL,AXRA+HOEND
0132  BA 0305                   +                   MOV     DX,AUXMD
0135  EE                        +                   OUT     DX,AL
0136  B0 40                     +                   MOV     AL,LON
0138  BA 0304                   +                   MOV     DX,ADRMD
013B  EE                        +                   OUT     DX,AL
013C  B0 08                     +                   MOV     AL,PON
013E  BA 0305                   +                   MOV     DX,AUXMD
0141  EE                        +                   OUT     DX,AL
                                                    OUT_GTSB
0142  B4 F6                     +                   MOV     AH,GTSB
0144  E8 0026 R                 +                   CALL    CMD8292         ;GoToStandBy
                                                    READ_AL
0147  BA 0301                   +                   MOV     DX,INT1
014A  EC                        + ??0008:           IN      AL,DX           ;get int 1 status
014B  24 01                     +                   AND     AL,BIM          ;check for byte in
014D  74 FB                     +                   JZ      ??0008          ;if not, try again
014F  BA 0300                   +                   MOV     DX,DIN
0152  EC                        +                   IN      AL,DX           ;get byte
                                                    READ_AL
0153  BA 0301                   +                   MOV     DX,INT1
```

```
0156  EC                   + ??0009:    IN      AL,DX          ;get int 1 status
0157  24 01                +            AND     AL,BIM         ;check for byte in
0159  74 FB                +            JZ      ??0009         ;if not, try again
015B  BA 0300              +            MOV     DX,DIN
015E  EC                   +            IN      AL,DX          ;get byte
015F  50                                PUSH    AX
                                        OUT_TCSY
0160  B4 FD                +            MOV     AH,TCSY
0162  E8 0026 R            +            CALL    CMD8292
0165  B0 80                +            MOV     AL,AXRA
0167  BA 0305              +            MOV     DX,AUXMD
016A  EE                   +            OUT     DX,AL
016B  B0 03                +            MOV     AL,FNHSK
016D  BA 0305              +            MOV     DX,AUXMD
0170  EE                   +            OUT     DX,AL
0171  B0 80                +            MOV     AL,TON
0173  BA 0304              +            MOV     DX,ADRMD
0176  EE                   +            OUT     DX,AL
0177  B0 00                +            MOV     AL,PON
0179  BA 0305              +            MOV     DX,AUXMD
017C  EE                   +            OUT     DX,AL
017D  58                                POP     AX
017E  24 02                             AND     AL,QflagMask
0180  74 9D                             JZ      HOLDHERE             ;repeat until Q<>0
                                                                     ;(TR8837 finished digitizing)
0182  C3                                RET
0183                         HLDFOR8837 ENDP
```

```
                                    PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;                                                                               :
                                    ;             P R O C E D U R E   G P I B   I N I T I A L I Z A T I O N         :
                                    ;                                                                               :
                                    ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;
                                    ;INPUTS:        none
                                    ;OUTPUTS:       none
                                    ;CALLS:         none
                                    ;DESTROYS:      nothing
                                    ;
 0183                               INIT            PROC    NEAR
 0183  50                                           PUSH    AX
 0184  52                                           PUSH    DX
 0185  9C                                           PUSHF
                                                    WAIT_COM                        ;wait for 92 IBF low
 0186  BA 0309                   +                  MOV     DX,INTST
 0189  EC                        + ??000A:          IN      AL,DX           ;get TCI status bit
 018A  24 02                     +                  AND     AL,IBFBT
 018C  75 FB                     +                  JNZ     ??000A          ;wait until false
                                                    OUTB    CMD92,RSET              ;reset 8292
 018E  B0 F2                     +                  MOV     AL,RSET
 0190  BA 0309                   +                  MOV     DX,CMD92
 0193  EE                        +                  OUT     DX,AL
                                                    OUTB    AUXMD,RESET91           ;     & 8291.
 0194  B0 02                     +                  MOV     AL,RESET91
 0196  BA 0305                   +                  MOV     DX,AUXMD
 0199  EE                        +                  OUT     DX,AL
                                                                                    ;(same as pulses to reset pins)
                                                    WAIT_COM                        ;wait for 92 IBF low
 019A  BA 0309                   +                  MOV     DX,INTST
 019D  EC                        + ??000B:          IN      AL,DX           ;get TCI status bit
 019E  24 02                     +                  AND     AL,IBFBT
 01A0  75 FB                     +                  JNZ     ??000B          ;wait until false
                                                    OUTB    INTMR,INTM              ;enable 8292 TCI
 01A2  B0 A0                     +                  MOV     AL,INTM
 01A4  BA 0308                   +                  MOV     DX,INTMR
 01A7  EE                        +                  OUT     DX,AL
                                                    OUTB    ADR01,DTDL1 ;MDA        ;disable major talker/listener
 01A8  B0 60                     +                  MOV     AL,DTDL1
 01AA  BA 0306                   +                  MOV     DX,ADR01
 01AD  EE                        +                  OUT     DX,AL
                                                    OUTB    ADR01,DTDL2             ;disable minor talker/listener
 01AE  B0 E0                     +                  MOV     AL,DTDL2
 01B0  BA 0306                   +                  MOV     DX,ADR01
 01B3  EE                        +                  OUT     DX,AL
                                                    OUTB    ADRMD,TON               ;Talk only mode
 01B4  B0 80                     +                  MOV     AL,TON
 01B6  BA 0304                   +                  MOV     DX,ADRMD
 01B9  EE                        +                  OUT     DX,AL
                                                    OUTB    AUXMD,CLKRT             ;5 MHz for delay timer
 01BA  B0 25                     +                  MOV     AL,CLKRT
 01BC  BA 0305                   +                  MOV     DX,AUXMD
 01BF  EE                        +                  OUT     DX,AL
                                                    OUTB    INT1,BOM+BIM            ;set byte in/out intr.s
```

```
01C0  B0 03                +                 MOV     AL,BOM+BIM
01C2  BA 0301              +                 MOV     DX,INT1
01C5  EE                   +                 OUT     DX,AL
                                             CLR     AL
01C6  32 C0                +                 XOR     AL,AL
01C8  BA 0302                                MOV     DX,INT2             ;disable some 8291 mask bits
01CB  EE                                     OUT     DX,AL
01CC  BA 0305                                MOV     DX,AUXMD            ;immediate execute PON
01CF  EE                                     OUT     DX,AL
01D0  BA 0019                                MOV     DX,25               ;this pause was added
01D3  4A                    PATCHPAUSE:      DEC     DX                  ;  as a patch for operation
01D4  75 FD                                  JNZ     PATCHPAUSE          ;               ( <<1ms )
01D6  9D                                     POPF                        ;restore stack
01D7  5A                                     POP     DX
01D8  5B                                     POP     AX
01D9  C3                                     RET     0
01DA                        INIT             ENDP
```

```
                                PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                ;                                                                               :
                                ;               C A M A C   C O M M A N D   P R O C E D U R E                   :
                                ;                                                                               :
                                ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                ;
                                ;INPUTS:        SEG:OFS pointer to a command string on stack before call
                                ;OUTPUTS:       none
                                ;CALLS:         CAMACMND
                                ;DESTROYS:      SI, ES & CAMACMND's registers
                                ;
                                ;NOTE:     This procedure is functionally identicall to CAMACMND except the
                                ;       string pointer ES:SI is loadded from the stack by this routine.
                                ;          It is used to call CAMACMND from turbo pascal where:
                                ;               TYPE
                                ;                  stringtype = STRING[?];
                                ;               PROCEDURE procname(VAR name:stringtype); EXTERNAL'CAMACOM.BIN';
                                ;         is used to access it. (lower case names supplied by user)
                                ;
01DA                            CAMACOM         PROC    NEAR
01DA  55                                        PUSH    BP
01DB  8B EC                                     MOV     BP,SP

01DD  C4 76 04                                  LES     SI, DWORD PTR SS:[BP+4] ;point to string
01E0  E8 003A R                                 CALL    CAMACMND                ; & procede with call

01E3  8B E5                                     MOV     SP,BP
01E5  5D                                        POP     BP
01E6  C2 0004                                   RET     4
01E9                            CAMACOM         ENDP
```

```
                                PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -:
                                ;                                                                       :
                                ;            B L O C K   A C Q U I R E   R O U T I N E                 :
                                ;                                                                       :
                                ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -:
                                ;
                                ;INPUTS:        Values on stack:
                                ;                       DWORD: pointer to array to be read
                                ;                       WORD : number of bytes to transfer
                                ;                       DWORD: pointer to 7 free bytes
                                ;OUTPUTS:       none
                                ;CALLS:         COM8292, CAMACMND, HLDFOR8837
                                ;DESTROYS:      ???
                                ;
                                ;from pascal:
                                ;   PROCEDURE xxxx(VAR ARY  : arytype;
                                ;                  NUMBYTES : INTEGER;
                                ;                   VAR DUMMY: STRING[6]  );
                                ;
01E9                            Get1Block       PROC    NEAR
01E9  55                                        PUSH    BP
01EA  8B EC                                     MOV     BP,SP
01EC  1E                                        PUSH    DS
01ED  9C                                        PUSHF           ;save interupt request status
01EE  FA                                        CLI             ;  ½  disable interupt requests
01EF  C4 76 04                                  LES     SI,DWORD PTR SS:[BP+4]  ;get pointer to a string
01F2  8B 5E 08                                  MOV     BX, WORD PTR SS:[BP+8]  ;number of bytes to transfer
01F5  C5 7E 0A                                  LDS     DI,DWORD PTR SS:[BP+10] ;get pointer to array location

01F8  E8 00FD R                                 CALL    HLDFOR8837

01FB  26: C6 04 03                              MOV     BYTE PTR ES:[SI+0],3            ;string size
01FF  26: C6 44 01 11                           MOV     BYTE PTR ES:[SI+1],CAMAC_FN+ ENBLREAD
0204  26: C6 44 02 00                           MOV     BYTE PTR ES:[SI+2],0            ;channel 00
0209  26: C6 44 03 0B                           MOV     BYTE PTR ES:[SI+3],Slot8837     ;camac slot # for 8837
020E  E8 00B3 R                                 CALL    CAMACMNDS                 ;execute camac cycle

0211  26: C6 04 01                              MOV     BYTE PTR ES:[SI+0],1            ;string size
0215  26: C6 44 01 69                           MOV     BYTE PTR ES:[SI+1],BYTERET+ HiSpeed+ Ret2Byte
021A  E8 00B3 R                                 CALL    CAMACMNDS                 ;execute camac cycle

021D  26: C6 04 01                              MOV     BYTE PTR ES:[SI+0],1            ;string size
0221  26: C6 44 01 02                           MOV     BYTE PTR ES:[SI+1],CAMAC_FN+ ReemDATA
0226  E8 00B3 R                                 CALL    CAMACMNDS                 ;execute camac cycle

                                                ;  G P I B   D M A   T R A N S F E R :
                                                OUT_MLA
0229  BA 0301                   +               MOV     DX,INT1
022C                            + ??000C:
022C  EC                        +               IN      AL,DX           ;get int 1 status
022D  24 02                     +               AND     AL,BOM          ;check for byte out
022F  74 FB                     +               JZ      ??000C          ;if not, try again
0231  B0 21                     +               MOV     AL,MLA
0233  BA 0300                   +               MOV     DX,DOUT
0236  EE                        +               OUT     DX,AL
```

```
0237  B0 82                  +           MOV     AL,AXRA+HOEND
0239  BA 0305                +           MOV     DX,AUXMD
023C  EE                     +           OUT     DX,AL
023D  B0 40                  +           MOV     AL,LON
023F  BA 0304                +           MOV     DX,ADRMD
0242  EE                     +           OUT     DX,AL
0243  B0 00                  +           MOV     AL,PON
0245  BA 0305                +           MOV     DX,AUXMD
0248  EE                     +           OUT     DX,AL
                                         OUT_GTSB                ;Go To Stand By
0249  B4 F6                  +           MOV     AH,GTSB
024B  E8 0026 R              +           CALL    CMD8292         ;GoToStandBy

                                         ;use DMA channel 3
                                         OUTB    PRTIDA,DMACM    ;configure DMA on tecmar board.
024E  B0 E0                  +           MOV     AL,DMACM
0250  BA 030C                +           MOV     DX,PRTIDA
0253  EE                     +           OUT     DX,AL
                                         OUTB    15,00AH         ;disable DMA channels 1 & 3
0254  B0 0A                  +           MOV     AL,00AH
0256  BA 000F                +           MOV     DX,15
0259  EE                     +           OUT     DX,AL
                                         OUTB    11,047H         ;set DMA channel 3 for mem write
025A  B0 47                  +           MOV     AL,047H
025C  BA 000B                +           MOV     DX,11
025F  EE                     +           OUT     DX,AL
0260  32 C0                              XOR     AL,AL
0262  E6 0C                              OUT     12,AL           ;set DMA 1st byte flip flop
0264  8B C7                              MOV     AX,DI
0266  E6 06                              OUT     6,AL            ;DMA write address, low  byte
0268  8A C4                              MOV     AL,AH
026A  E6 06                              OUT     6,AL            ; "    "       "     high  "
026C  8C D8                              MOV     AX,DS           ;set DMA write page (64K/page)
026E  B1 04                              MOV     CL,4     ;ASSUMES SEG = ABSOLUTE 64K BOUNDRY VALUE
0270  D3 C0                              ROL     AX,CL    ; eg) DS= n*$1000 ,n is positive integer
0272  E6 82                              OUT     082H,AL

0274  8A C3                              MOV     AL,BL           ;MEM SIZE
0276  E6 07                              OUT     7,AL
0278  8A C7                              MOV     AL,BH
027A  E6 07                              OUT     7,AL
                                         OUTB    INT2,010H       ;enable 8191A for DMA
027C  B0 10                  +           MOV     AL,010H
027E  BA 0302                +           MOV     DX,INT2
0281  EE                     +           OUT     DX,AL
                                         OUTB    AUXMD,PON
0282  B0 00                  +           MOV     AL,PON
0284  BA 0305                +           MOV     DX,AUXMD
0287  EE                     +           OUT     DX,AL
0288  32 C0                              XOR     AL,AL           ;enable all channels for DMA
028A  E6 0F                              OUT     15,AL
028C  E4 08                  HLDFRTC:    IN      AL,8            ;wait for TC flag
028E  24 08                              AND     AL,8
0290  74 FA                              JZ      HLDFRTC
                                         OUTB    PRTIDA,0;un-configure DMA on tecmar board.
```

```
0292  B0 00                 +               MOV     AL,0
0294  BA 030C               +               MOV     DX,PRTIDA
0297  EE                    +               OUT     DX,AL

                                            T_CONTROL_SYN          ;400 ous to control mode
0298  B4 FD                 +               MOV     AH,TCSY
029A  E8 0026 R             +               CALL    CMD8292
029D  B0 80                 +               MOV     AL,AXRA
029F  BA 0305               +               MOV     DX,AUXMD
02A2  EE                    +               OUT     DX,AL
02A3  B0 03                 +               MOV     AL,FNHSK
02A5  BA 0305               +               MOV     DX,AUXMD
02A8  EE                    +               OUT     DX,AL
02A9  B0 80                 +               MOV     AL,TON
02AB  BA 0304               +               MOV     DX,ADRMD
02AE  EE                    +               OUT     DX,AL
02AF  B0 00                 +               MOV     AL,PON
02B1  BA 0305               +               MOV     DX,AUXMD
02B4  EE                    +               OUT     DX,AL

02B5  26: C6 04 01                          MOV     BYTE PTR ES:[SI+0],1            ;string size
02B9  26: C6 44 01 09                       MOV     BYTE PTR ES:[SI+1],CAMAC_FN+ StartDigt
02BE  E8 003A R                             CALL    CAMACMND                ;execute camac cycle

02C1  9D                                    POPF            ;re_enable interupt requests
02C2  1F                                    POP     DS
02C3  8B E5                                 MOV     SP,BP
02C5  5D                                    POP     BP
02C6  C2 000A                               RET     10
02C9                          Get1Block     ENDP


02C9                          CSEG    ENDS
                                      END     START
```

```
GPIB-CAMAC OPERATIONS


Macros:

                N a m e                 Length

CLR. . . . . . . . . . . . . . .        0001
LISTEN . . . . . . . . . . . . .        0007
LISTEN_FST . . . . . . . . . . .        0008
OUTB . . . . . . . . . . . . . .        0002
OUT_GTSB . . . . . . . . . . . .        0002
OUT_MLA. . . . . . . . . . . . .        0006
OUT_MTA. . . . . . . . . . . . .        0003
OUT_TCSY . . . . . . . . . . . .        0006
OUT_TCSY_FST . . . . . . . . . .        0006
READ_AL. . . . . . . . . . . . .        0003
SETF . . . . . . . . . . . . . .        0001
TALK . . . . . . . . . . . . . .        0005
T_CONTROL_SYN. . . . . . . . . .        0002
UNLISTEN . . . . . . . . . . . .        0003
WAITI. . . . . . . . . . . . . .        0004
WAITO. . . . . . . . . . . . . .        0004
WAITT. . . . . . . . . . . . . .        0003
WAIT_COM . . . . . . . . . . . .        0004
WRITE_AL . . . . . . . . . . . .        0003

Segments and groups:

                N a m e                 Size    align   combine class

CSEG . . . . . . . . . . . . . .        02C9    PARA    NONE

Symbols:

                N a m e                 Type    Value   Attr

ABORT. . . . . . . . . . . . . .        Number  00F9
ADR0I. . . . . . . . . . . . . .        Number  0306
ADRMD. . . . . . . . . . . . . .        Number  0304
ADRST. . . . . . . . . . . . . .        Number  0304
AUXMD. . . . . . . . . . . . . .        Number  0305
AXRA . . . . . . . . . . . . . .        Number  0080
AXRB . . . . . . . . . . . . . .        Number  00A0
BEGIFTHEN. . . . . . . . . . . .        L NEAR  0054    CSEG
BEGIFTHENS . . . . . . . . . . .        L NEAR  00C2    CSEG
BIM. . . . . . . . . . . . . . .        Number  0001
BOICM. . . . . . . . . . . . . .        Number  0010
BOM. . . . . . . . . . . . . . .        Number  0002
BUSST. . . . . . . . . . . . . .        Number  0308
BYTERET. . . . . . . . . . . . .        Number  0060
CAMCY. . . . . . . . . . . . . .        Number  0003
CAMACMND . . . . . . . . . . . .        N PROC  003A    CSEG    Length =0079
CAMACMNDS. . . . . . . . . . . .        N PROC  00B3    CSEG    Length =004A
CAMACOM. . . . . . . . . . . . .        N PROC  01DA    CSEG    Length =000F
CAMAC_FLAGS. . . . . . . . . . .        Number  0020
CAMAC_FN . . . . . . . . . . . .        Number  0000
CFLAGMASK. . . . . . . . . . . .        Number  0002
```

```
The IBM Personal Computer MACRO Assembler 02-02-87      PAGE    Symbols-2
GPIB-CAMAC OPERATIONS


CLKRT. . . . . . . . . . . . . .        Number  0025
CLRLAM . . . . . . . . . . . . .        Number  000A
CLRST. . . . . . . . . . . . . .        Number  0308
CMD8292. . . . . . . . . . . . .        N PROC  0026    CSEG    Length =0014
CMD8292S . . . . . . . . . . . .        N PROC  0009    CSEG    Length =001D
CMD92. . . . . . . . . . . . . .        Number  0309
CPT. . . . . . . . . . . . . . .        Number  0080
CPTEN. . . . . . . . . . . . . .        Number  0001
CPTRG. . . . . . . . . . . . . .        Number  0305
CRATE_ADDR . . . . . . . . . . .        Number  000A
DEFLT. . . . . . . . . . . . . .        Number  0000
DIN. . . . . . . . . . . . . . .        Number  0300
DMACM. . . . . . . . . . . . . .        Number  0080
DOUT . . . . . . . . . . . . . .        Number  0300
DSBLLAM. . . . . . . . . . . . .        Number  0018
DTDL1. . . . . . . . . . . . . .        Number  0060
DTDL2. . . . . . . . . . . . . .        Number  00E0
EDEOS. . . . . . . . . . . . . .        Number  0004
ENBLDISP . . . . . . . . . . . .        Number  0012
ENBLLAM. . . . . . . . . . . . .        Number  001A
ENBLREAD . . . . . . . . . . . .        Number  0011
ENDIFTHEN. . . . . . . . . . . .        L NEAR  00A0    CSEG
ENDIFTHENS . . . . . . . . . . .        L NEAR  00F4    CSEG
ENDMK. . . . . . . . . . . . . .        Number  0010
EOIS . . . . . . . . . . . . . .        Number  0008
EOIST. . . . . . . . . . . . . .        Number  0020
EOSR . . . . . . . . . . . . . .        Number  0307
ERFLG. . . . . . . . . . . . . .        Number  0308
ERRBIT . . . . . . . . . . . . .        Number  0040
ERRM . . . . . . . . . . . . . .        Number  0308
EVBIT. . . . . . . . . . . . . .        Number  0010
EVREG. . . . . . . . . . . . . .        Number  0308
EXPP . . . . . . . . . . . . . .        Number  00F5
FILTERED . . . . . . . . . . . .        Number  0080
FNHSK. . . . . . . . . . . . . .        Number  0003
GET. . . . . . . . . . . . . . .        Number  0008
GET1BLOCK. . . . . . . . . . . .        N PROC  01E9    CSEG    Length =00E0
GIDL . . . . . . . . . . . . . .        Number  00F1
GROUND . . . . . . . . . . . . .        Number  0040
GSEC . . . . . . . . . . . . . .        Number  00F4
GTSB . . . . . . . . . . . . . .        Number  00F6
HISPEED. . . . . . . . . . . . .        Number  0008
HLDFOR8837 . . . . . . . . . . .        N PROC  00FD    CSEG    Length =0086
HLDFRTC. . . . . . . . . . . . .        L NEAR  028C    CSEG
HOEND. . . . . . . . . . . . . .        Number  0002
HOLDHERE . . . . . . . . . . . .        L NEAR  011F    CSEG
HOSHK. . . . . . . . . . . . . .        Number  0001
IACK . . . . . . . . . . . . . .        Number  000B
IBFBT. . . . . . . . . . . . . .        Number  0002
IBFICM . . . . . . . . . . . . .        Number  0008
IFCRBIT. . . . . . . . . . . . .        Number  0004
IFLAGMASK. . . . . . . . . . . .        Number  0008
INIT . . . . . . . . . . . . . .        N PROC  0183    CSEG    Length =0057
INT1 . . . . . . . . . . . . . .        Number  0301
INT2 . . . . . . . . . . . . . .        Number  0302
```

```
INTM . . . . . . . . . . . . . .        Number  00A0
INTM1. . . . . . . . . . . . . .        Number  0301
INTMR. . . . . . . . . . . . . .        Number  0308
INTST. . . . . . . . . . . . . .        Number  0309
LA . . . . . . . . . . . . . . .        Number  0001
LAMENBL. . . . . . . . . . . . .        Number  0001
LON. . . . . . . . . . . . . . .        Number  0040
MDA. . . . . . . . . . . . . . .        Number  0001
MLA. . . . . . . . . . . . . . .        Number  0021
MODE1. . . . . . . . . . . . . .        Number  0001
MTA. . . . . . . . . . . . . . .        Number  0041
NOQENBL. . . . . . . . . . . . .        Number  0002
NOXENBL. . . . . . . . . . . . .        Number  0004
NVCMD. . . . . . . . . . . . . .        Number  0007
OBFBIT . . . . . . . . . . . . .        Number  0001
OBFICM . . . . . . . . . . . . .        Number  0004
OUTDATA. . . . . . . . . . . . .        L NEAR  006F    CSEG
OUTDATAS . . . . . . . . . . . .        L NEAR  00D0    CSEG
PATCHPAUSE . . . . . . . . . . .        L NEAR  01D3    CSEG
PON. . . . . . . . . . . . . . .        Number  0000
PPC. . . . . . . . . . . . . . .        Number  0005
PPD. . . . . . . . . . . . . . .        Number  0070
PPE. . . . . . . . . . . . . . .        Number  0060
PPU. . . . . . . . . . . . . . .        Number  0015
PRT91. . . . . . . . . . . . . .        Number  0300
PRT92. . . . . . . . . . . . . .        Number  0308
PRTIDA . . . . . . . . . . . . .        Number  030C
QFLAGMASK. . . . . . . . . . . .        Number  0002
RBST . . . . . . . . . . . . . .        Number  00E7
RCST . . . . . . . . . . . . . .        Number  00E6
READGAIN . . . . . . . . . . . .        Number  0000
READOFS. . . . . . . . . . . . .        Number  0001
RERF . . . . . . . . . . . . . .        Number  00E4
RERM . . . . . . . . . . . . . .        Number  00EA
RESET91. . . . . . . . . . . . .        Number  0002
RET2BYTE . . . . . . . . . . . .        Number  0001
RET3BYTE . . . . . . . . . . . .        Number  0002
RET4BYTE . . . . . . . . . . . .        Number  0004
REVC . . . . . . . . . . . . . .        Number  00E3
RINM . . . . . . . . . . . . . .        Number  00E5
RMEMDATA . . . . . . . . . . . .        Number  0002
RMODULE. . . . . . . . . . . . .        Number  0003
RPARMS . . . . . . . . . . . . .        Number  0000
RSAMPLE. . . . . . . . . . . . .        Number  0001
RSET . . . . . . . . . . . . . .        Number  00F2
RSTI . . . . . . . . . . . . . .        Number  00F3
RTOUT. . . . . . . . . . . . . .        Number  00E9
SDC. . . . . . . . . . . . . . .        Number  0004
SDEOI. . . . . . . . . . . . . .        Number  0006
SETGAIN. . . . . . . . . . . . .        Number  0010
SETOFS . . . . . . . . . . . . .        Number  0011
SLOC . . . . . . . . . . . . . .        Number  00F7
SLOT8100 . . . . . . . . . . . .        Number  0005
SLOT8837 . . . . . . . . . . . .        Number  000B
SPCNI. . . . . . . . . . . . . .        Number  00F0
```

```
SPD. . . . . . . . . . . . . . .        Number  0019
SPE. . . . . . . . . . . . . . .        Number  0018
SPICM. . . . . . . . . . . . . .        Number  0020
SREM . . . . . . . . . . . . . .        Number  00FB
SRQBT. . . . . . . . . . . . . .        Number  0020
SRQCNTRL . . . . . . . . . . . .        Number  0040
SSDIGTIZE. . . . . . . . . . . .        Number  0008
START. . . . . . . . . . . . . .        L NEAR  0000    CSEG
STARTDIGT. . . . . . . . . . . .        Number  0009
STCNI. . . . . . . . . . . . . .        Number  00FE
STOPDIGT . . . . . . . . . . . .        Number  0019
SYCBIT . . . . . . . . . . . . .        Number  0080
TA . . . . . . . . . . . . . . .        Number  0002
TCASY. . . . . . . . . . . . . .        Number  00FC
TCICM. . . . . . . . . . . . . .        Number  0002
TCILC. . . . . . . . . . . . . .        Number  030A
TCIM . . . . . . . . . . . . . .        Number  0001
TCNTR. . . . . . . . . . . . . .        Number  00FA
TCSY . . . . . . . . . . . . . .        Number  00FD
TCT. . . . . . . . . . . . . . .        Number  0009
TON. . . . . . . . . . . . . . .        Number  0060
TOREG. . . . . . . . . . . . . .        Number  0308
TOUT1. . . . . . . . . . . . . .        Number  0001
TOUT2. . . . . . . . . . . . . .        Number  0002
TOUT3. . . . . . . . . . . . . .        Number  0004
TSTLAM . . . . . . . . . . . . .        Number  001B
TSTMLAM. . . . . . . . . . . . .        Number  0008
UNFILTERED . . . . . . . . . . .        Number  0000
UNGROUND . . . . . . . . . . . .        Number  0000
UNL. . . . . . . . . . . . . . .        Number  003F
VSCMD. . . . . . . . . . . . . .        Number  000F
WEVC . . . . . . . . . . . . . .        Number  00E2
WOUT . . . . . . . . . . . . . .        Number  00E1
WPARMS . . . . . . . . . . . . .        Number  0010
XFLAGMASK. . . . . . . . . . . .        Number  0001
ZFLAGMASK. . . . . . . . . . . .        Number  0001
??0000 . . . . . . . . . . . . .        L NEAR  000C    CSEG
??0001 . . . . . . . . . . . . .        L NEAR  0020    CSEG
??0002 . . . . . . . . . . . . .        L NEAR  003F    CSEG
??0003 . . . . . . . . . . . . .        L NEAR  0057    CSEG
??0004 . . . . . . . . . . . . .        L NEAR  0065    CSEG
??0005 . . . . . . . . . . . . .        L NEAR  007A    CSEG
??0006 . . . . . . . . . . . . .        L NEAR  00A3    CSEG
??0007 . . . . . . . . . . . . .        L NEAR  0125    CSEG
??0008 . . . . . . . . . . . . .        L NEAR  014A    CSEG
??0009 . . . . . . . . . . . . .        L NEAR  0156    CSEG
??000A . . . . . . . . . . . . .        L NEAR  0189    CSEG
??000B . . . . . . . . . . . . .        L NEAR  019D    CSEG
??000C . . . . . . . . . . . . .        L NEAR  022C    CSEG

Warning Severe
Errors  Errors
0       0
```

```
                              .list
```

```
                                   PAGE
0000                               CSEG    SEGMENT PUBLIC
                                           ASSUME  CS:CSEG
0000                               START:
                                   ;               V E C T O R   T A B L E :
0000  E9 0125 R                                    JMP     INIT
0003  E9 017C R                                    JMP     CAMACOM
0006  E9 0188 R                                    JMP     Get1block
                                   ;
                                   ;               P R O G R A M   E Q U A T E S :
                                   ;
= 000A                             CRATE_ADDR      EQU     10      ;GPIB address of camac crate
```

```
                                PAGE;-----------------------------------------;
                                ;                                               :
                                :            P R O C E D U R E   C O M M A N D 8 2 9 2    :
                                ;                                               :
                                ;-----------------------------------------;
                                ;
                                ;INPUTS:        command byte in AH
                                ;OUTPUTS:       none
                                ;CALLS:         none
                                ;DESTROYS:      AL, DX & Flags
                                ;
0009                            CMD8292         PROC    NEAR
                                                WAIT_COM                ;wait until 8292 ready for next command
0009  BA 0339               +                   MOV     DX,INTST
000C  EC                    + ??0000:           IN      AL,DX           ;get TCI status bit
000D  24 02                 +                   AND     AL,IBFBT
000F  75 FB                 +                   JNZ     ??0000          ;wait until false
                                                OUTB    TCILC,0         ;Clear TCI Latch
0011  B0 00                 +                   MOV     AL,0
0013  BA 033A               +                   MOV     DX,TCILC
0016  EE                    +                   OUT     DX,AL
0017  BA 0339                                   MOV     DX,CMD92
001A  8A C4                                     MOV     AL,AH           ;get command byte
001C  EE                                        OUT     DX,AL           ;& output it to 8292
                                                WAITT
001D  BA 033C               +                   MOV     DX,PRTIDA
0020  EC                    + ??0001:           IN      AL,DX           ;get TCI status bit
0021  24 01                 +                   AND     AL,TCIM
0023  74 FB                 +                   JZ      ??0001          ;wait until false
0025  C3                                        RET
0026                            CMD8292         ENDP
```

```
                                PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                ;                                                                      :
                                ;               P R O C E D U R E   C A M A C   C O M M A N D            :
                                ;                                                                      :
                                ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                ;
                                ;INPUTS:        ES:SI points to command string
                                ;OUTPUTS:       none
                                ;CALLS:         COM8292
                                ;DESTROYS:      AX, DX & Flags
                                ;
0026                            CAMACMND        PROC    NEAR
0026  51                                        PUSH    CX
0027  56                                        PUSH    SI
                                                LISTEN_FST CRATE_ADDR   ;tell crate to listen
0028  BA 0331                +                  MOV     DX,INT1
002B                         + ??0002:
002B  EC                     +                  IN      AL,DX           ;get int 1 status
002C  24 02                  +                  AND     AL,BOM          ;check for byte out
002E  74 FB                  +                  JZ      ??0002          ;if not, try again
0030  B0 0A                  +                  MOV     AL,CRATE_ADDR
0032  04 20                  +                  ADD     AL,020H
0034  BA 0330                +                  MOV     DX,DOUT
0037  EE                     +                  OUT     DX,AL           ;& output to 8291
0038  26: 8A 0C                                 MOV     CL,ES:[SI]      ; & get string size
003B  80 F9 00                                  CMP     CL,0
003E  74 4C                                     JE      EndIfThen       ;send no data if nul string
0040                            BegIfThen:      OUT_MTA                 ;send my talk address
0040  BA 0331                +                  MOV     DX,INT1
0043                         + ??0003:
0043  EC                     +                  IN      AL,DX           ;get int 1 status
0044  24 02                  +                  AND     AL,BOM          ;check for byte out
0046  74 FB                  +                  JZ      ??0003          ;if not, try again
0048  B0 41                  +                  MOV     AL,MTA
004A  BA 0330                +                  MOV     DX,DOUT
004D  EE                     +                  OUT     DX,AL
004E  BA 0331                +                  MOV     DX,INT1
0051                         + ??0004:
0051  EC                     +                  IN      AL,DX           ;get int 1 status
0052  24 02                  +                  AND     AL,BOM          ;check for byte out
0054  74 FB                  +                  JZ      ??0004          ;if not, try again
                                                OUT_GTSB                ;go to stand by
0056  B4 F6                  +                  MOV     AH,GTSB
0058  E8 0009 R              +                  CALL    CMD8292         ;GoToStandBy
005B  46                        OUTDATA:        INC     SI
005C  26: 8A 04                                 MOV     AL,ES:[SI]      ;send the data
                                                WRITE_AL
005F  BA 0330                +                  MOV     DX,DOUT
0062  EE                     +                  OUT     DX,AL           ;& output to 8291
0063  BA 0331                +                  MOV     DX,INT1
0066                         + ??0005:
0066  EC                     +                  IN      AL,DX           ;get int 1 status
0067  24 02                  +                  AND     AL,BOM          ;check for byte out
0069  74 FB                  +                  JZ      ??0005          ;if not, try again
006B  FE C9                                     DEC     CL
```

```
006D  75 EC                            JNZ     OUTDATA
                                       OUT_TCSY                  ;take control syncronously
006F  B4 FD               +            MOV     AH,TCSY
0071  E8 0009 R           +            CALL    CMD8292
0074  B0 80               +            MOV     AL,AXRA
0076  BA 0335             +            MOV     DX,AUXMD
0079  EE                  +            OUT     DX,AL
007A  B0 03               +            MOV     AL,FNHSK
007C  BA 0335             +            MOV     DX,AUXMD
007F  EE                  +            OUT     DX,AL
0080  B0 90               +            MOV     AL,TON
0082  BA 0334             +            MOV     DX,ADRMD
0085  EE                  +            OUT     DX,AL
0086  B0 00               +            MOV     AL,PON
0088  BA 0335             +            MOV     DX,AUXMD
008B  EE                  +            OUT     DX,AL
008C                        EndIfThen: TALK    CRATE_ADDR
008C  BA 0331             +            MOV     DX,INT1
008F                      + ??0006:
008F  EC                  +            IN      AL,DX         ;get int 1 status
0090  24 02               +            AND     AL,BOM        ;check for byte out
0092  74 FB               +            JZ      ??0006        ;if not, try again
0094  B0 0A               +            MOV     AL,CRATE_ADDR ;get address byte
0096  04 40               +            ADD     AL,040H
0098  BA 0330             +            MOV     DX,DOUT
009B  EE                  +            OUT     DX,AL         ;& output it to 8292
009C  5E                               POP     SI
009D  59                               POP     CX
009E  C3                               RET
009F                        CAMACMND   ENDP
```

```
                                  PAGE;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                  ;                                                                        :
                                  ;           H O L D   W H I L E   T R 8 8 3 7   D I G I T I Z I N G      :
                                  ;                                                                        :
                                  ;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                  ;
                                  ;INPUTS:        none
                                  ;OUTPUTS:       none
                                  ;CALLS:         CMD8292, CAMACMND
                                  ;DESTROYS:      AX, ???
                                  ;
009F                              HLDFGR8837      PROC    NEAR
009F  26: C6 04 03                                MOV     BYTE PTR ES:[SI],3       ;command string size
00A3  26: C6 44 01 19                             MOV     BYTE PTR ES:[SI+1],TstLAM ;function=test lam via Q flag
00A8  26: C6 44 02 00                             MOV     BYTE PTR ES:[SI+2],0             ;channel #0
00AD  26: C6 44 03 06                             MOV     BYTE PTR ES:[SI+3],TR8837slot    ;camac slot # for 8837
00B2  E8 0026 R                                   CALL    CAMACMND

00B5  26: C6 04 01                                MOV     BYTE PTR ES:[SI],1       ;command string size
00B9  26: C6 44 01 61                             MOV     BYTE PTR ES:[SI+1],ByteRet + Ret2Byte
00BE  E8 0026 R                                   CALL    CAMACMND

                                  ;               MOV     BYTE PTR ES:[SI],0       ;command string size
00C1                              HOLDHERE:
00C1  E8 0026 R                                   CALL    CAMACMND                 ;execute camac cycle
                                                  OUT_MLA                          ; & read two bytes
00C4  BA 0331                   +                 MOV     DX,INT1
00C7                            + ??0007:
00C7  EC                        +                 IN      AL,DX           ;get int 1 status
00C8  24 02                     +                 AND     AL,BOM          ;check for byte out
00CA  74 FB                     +                 JZ      ??0007          ;if not, try again
00CC  B0 21                     +                 MOV     AL,MLA
00CE  BA 0330                   +                 MOV     DX,DOUT
00D1  EE                        +                 OUT     DX,AL
00D2  B0 82                     +                 MOV     AL,AXRA+HOEND
00D4  BA 0335                   +                 MOV     DX,AUXMD
00D7  EE                        +                 OUT     DX,AL
00D8  B0 40                     +                 MOV     AL,LON
00DA  BA 0334                   +                 MOV     DX,ADRMD
00DD  EE                        +                 OUT     DX,AL
00DE  B0 00                     +                 MOV     AL,PON
00E0  BA 0335                   +                 MOV     DX,AUXMD
00E3  EE                        +                 OUT     DX,AL
                                                  OUT_GTSB
00E4  B4 F6                     +                 MOV     AH,GTSB
00E6  E8 0009 R                 +                 CALL    CMD8292         ;GoToStandBy
                                                  READ_AL
00E9  BA 0331                   +                 MOV     DX,INT1
00EC  EC                        + ??0008:         IN      AL,DX           ;get int 1 status
00ED  24 01                     +                 AND     AL,BIM          ;check for byte in
00EF  74 FB                     +                 JZ      ??0008          ;if not, try again
00F1  BA 0330                   +                 MOV     DX,DIN
00F4  EC                        +                 IN      AL,DX           ;get byte
                                                  READ_AL
00F5  BA 0331                   +                 MOV     DX,INT1
```

```
00F8  EC                    + ??0009:        IN      AL,DX           ;get int 1 status
00F9  24 01                 +                AND     AL,BIM          ;check for byte in
00FB  74 FB                 +                JZ      ??0009          ;if not, try again
00FD  BA 0330               +                MOV     DX,DIN
0100  EC                    +                IN      AL,DX           ;get byte
0101  50                                     PUSH    AX
                                             OUT_TCSY
0102  B4 FD                 +                MOV     AH,TCSY
0104  E8 0009 R             +                CALL    CMD8292
0107  B0 80                 +                MOV     AL,AXRA
0109  BA 0335               +                MOV     DX,AUXMD
010C  EE                    +                OUT     DX,AL
010D  B0 03                 +                MOV     AL,FNHSK
010F  BA 0335               +                MOV     DX,AUXMD
0112  EE                    +                OUT     DX,AL
0113  B0 80                 +                MOV     AL,TON
0115  BA 0334               +                MOV     DX,ADRMD
0118  EE                    +                OUT     DX,AL
0119  B0 00                 +                MOV     AL,PON
011B  BA 0335               +                MOV     DX,AUXMD
011E  EE                    +                OUT     DX,AL
011F  58                                     POP     AX
0120  24 02                                  AND     AL,QflagMask
0122  74 9D                                  JZ      HOLDHERE              ;repeat until Q<>0
                                                                           ;(TR8837 finished digitizing)
0124  C3                                     RET
0125                          HLDFOR8837     ENDP
```

```
0162  B0 03              +                 MOV    AL,BOM+BIM
0164  BA 0331            +                 MOV    DX,INT1
0167  EE                 +                 OUT    DX,AL
                                           CLR    AL
0168  32 C0              +                 XOR    AL,AL
016A  BA 0332                              MOV    DX,INT2              ;disable some 8291 mask bits
016D  EE                                   OUT    DX,AL
016E  BA 0335                              MOV    DX,AUXMD             ;immediate execute PON
0171  EE                                   OUT    DX,AL
0172  BA 0019                              MOV    DX,25                ;this pause was added
0175  4A                   PATCHPAUSE:     DEC    DX                   ;  as a patch for operation
0176  75 FD                                JNZ    PATCHPAUSE           ;               ( <<1ms )
0178  9D                                   POPF                        ;restore stack
0179  5A                                   POP    DX
017A  58                                   POP    AX
017B  C3                                   RET    0
017C                       INIT            ENDP
```

```
                              PAGE;-----------------------------------------------------;
                              ;                                                     :
                              ;             P R O C E D U R E   G P I B   I N I T I A L I Z A T I O N     :
                              ;                                                     :
                              ;-----------------------------------------------------;
                              ;
                              ;INPUTS:        none
                              ;OUTPUTS:       none
                              ;CALLS:         none
                              ;DESTROYS:      nothing
                              ;
0125                          INIT            PROC    NEAR
0125  50                                      PUSH    AX
0126  52                                      PUSH    DX
0127  9C                                      PUSHF
                                              WAIT_COM                ;wait for 92 IBF low
0128  BA 0339            +                    MOV     DX,INTST
012B  EC                 + ??000A:            IN      AL,DX        ;get TCI status bit
012C  24 02              +                    AND     AL,IBFBT
012E  75 FB              +                    JNZ     ??000A       ;wait until false
                                              OUTB    CMD92,RSET              ;reset 8292
0130  B0 F2              +                    MOV     AL,RSET
0132  BA 0339            +                    MOV     DX,CMD92
0135  EE                 +                    OUT     DX,AL
                                              OUTB    AUXMD,RESET91           ;    & 8291.
0136  B0 02              +                    MOV     AL,RESET91
0138  BA 0335            +                    MOV     DX,AUXMD
013B  EE                 +                    OUT     DX,AL
                                                                      ;(same as pulses to reset pins)
                                              WAIT_COM                ;wait for 92 IBF low
013C  BA 0339            +                    MOV     DX,INTST
013F  EC                 + ??000B:            IN      AL,DX        ;get TCI status bit
0140  24 02              +                    AND     AL,IBFBT
0142  75 FB              +                    JNZ     ??000B       ;wait until false
                                              OUTB    INTMR,INTM              ;enable 8292 TCI
0144  B0 A0              +                    MOV     AL,INTM
0146  BA 0338            +                    MOV     DX,INTMR
0149  EE                 +                    OUT     DX,AL
                                              OUTB    ADR01,DTDL1 ;MDA        ;disable major talker/listener
014A  B0 60              +                    MOV     AL,DTDL1
014C  BA 0336            +                    MOV     DX,ADR01
014F  EE                 +                    OUT     DX,AL
                                              OUTB    ADR01,DTDL2             ;disable minor talker/listener
0150  B0 E0              +                    MOV     AL,DTDL2
0152  BA 0336            +                    MOV     DX,ADR01
0155  EE                 +                    OUT     DX,AL
                                              OUTB    ADRMD,TON               ;Talk only mode
0156  B0 80              +                    MOV     AL,TON
0158  BA 0334            +                    MOV     DX,ADRMD
015B  EE                 +                    OUT     DX,AL
                                              OUTB    AUXMD,CLKRT             ;5 MHz for delay timer
015C  B0 25              +                    MOV     AL,CLKRT
015E  BA 0335            +                    MOV     DX,AUXMD
0161  EE                 +                    OUT     DX,AL
                                              OUTB    INT1,BOM+BIM            ;set byte in/out intr.s
```

```
                                    PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;                                                                    :
                                    ;             C A M A C   C O M M A N D   P R O C E D U R E         :
                                    ;                                                                    :
                                    ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                                    ;
                                    ;INPUTS:        SEG:OFS pointer to a command string on stack before call
                                    ;OUTPUTS:       none
                                    ;CALLS:         CAMACMND
                                    ;DESTROYS:      SI, ES & CAMACMND's registers
                                    ;
                                    ;NOTE:     This procedure is functionally identicall to CAMACMND except the
                                    ;       string pointer ES:SI is loaded from the stack by this routine.
                                    ;          It is used to call CAMACMND from turbo pascal where:
                                    ;               TYPE
                                    ;                  stringtype = STRING[?];
                                    ;               PROCEDURE procname(VAR name:stringtype); EXTERNAL'CAMACOM.BIN';
                                    ;         is used to access it. (lower case names supplied by user)
                                    ;
017C                                CAMACOM         PROC    NEAR
017C  55                                            PUSH    BP
017D  8B EC                                         MOV     BP,SP

017F  C4 76 04                                      LES     SI, DWORD PTR SS:[BP+4] ;point to string
0182  E8 0026 R                                     CALL    CAMACMND                ; & procede with call

0185  8B E5                                         MOV     SP,BP
0187  5D                                            POP     BP
0188  C2 0004                                       RET     4
018B                                CAMACOM         ENDP
```

```
                         PAGE; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                         ;                                                               :
                         ;           B L O C K   A Q U I R E   R O U T I N E             :
                         ;                                                               :
                         ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - :
                         ;
                         ;INPUTS:     Values on stack:
                         ;                  DWORD: pointer to array to be read
                         ;                  WORD : number of bytes to transfer
                         ;                  DWORD: pointer to 7 free bytes
                         ;OUTPUTS:     none
                         ;CALLS:       COM8292, CAMACMND, HLDFOR8837, DMA$XFER
                         ;DESTROYS:    ???
                         ;
                         EXTRN        DMA$XFER:NEAR
                         ;
                         ;from pascal:
                         ;   PROCEDURE xxxx(VAR ARY  : arytype;
                         ;                 NUMBYTES : INTEGER;
                         ;                 VAR DUMMY: STRING[6]  );
                         ;
018B                     Get1Block    PROC   NEAR
018B  55                              PUSH   BP
018C  8B EC                           MOV    BP,SP
018E  1E                              PUSH   DS
018F  9C                              PUSHF         ;save interupt request status
0190  FA                              CLI           ; &  disable interupt requests
0191  C4 76 04                        LES    SI,DWORD PTR SS:[BP+4]  ;get pointer to a string
0194  8B 5E 08                        MOV    BX, WORD PTR SS:[BP+8]  ;number of bytes to transfer
0197  C5 7E 0A                        LDS    DI,DWORD PTR SS:[BP+10] ;get pointer to array location

019A  E8 009F R                       CALL   HLDFOR8837

                                      ; assume positions 2 & 3 set by HLDFOR8837
019D  26: C6 04 01                    MOV    BYTE PTR ES:[SI+0],1          ;string size
01A1  26: C6 44 01 11                 MOV    BYTE PTR ES:[SI+1],CAMAC_FN+ ENBLREAD
01A6  E8 0026 R                       CALL   CAMACMND                ;execute camac cycle

01A9  26: C6 04 01                    MOV    BYTE PTR ES:[SI+0],1          ;string size
01AD  26: C6 44 01 69                 MOV    BYTE PTR ES:[SI+1],BYTERET+ HiSpeed+ Ret2Byte
01B2  E8 0026 R                       CALL   CAMACMND                ;execute camac cycle

01B5  26: C6 04 01                    MOV    BYTE PTR ES:[SI+0],1          ;string size
01B9  26: C6 44 01 02                 MOV    BYTE PTR ES:[SI+1],CAMAC_FN+ RmemDATA
01BE  E8 0026 R                       CALL   CAMACMND                ;execute camac cycle

                                      ;  G P I B   D M A   T R A N S F E R :
                                      OUT_MLA
01C1  BA 0331            +            MOV    DX,INT1
01C4                     + ??000C:
01C4  EC                 +            IN     AL,DX          ;get int 1 status
01C5  24 02              +            AND    AL,BOM         ;check for byte out
01C7  74 FB              +            JZ     ??000C         ;if not, try again
01C9  B0 21              +            MOV    AL,MLA
01CB  BA 0330            +            MOV    DX,DOUT
```

```
01CE  EE                    +               OUT     DX,AL
01CF  B0 82                 +               MOV     AL,AXRA+HOEND
01D1  BA 0335               +               MOV     DX,AUXMD
01D4  EE                    +               OUT     DX,AL
01D5  B0 40                 +               MOV     AL,LON
01D7  BA 0334               +               MOV     DX,ADRMD
01DA  EE                    +               OUT     DX,AL
01DB  B0 00                 +               MOV     AL,PON
01DD  BA 0335               +               MOV     DX,AUXMD
01E0  EE                    +               OUT     DX,AL
                                            OUT_GTSB                ;Go To Stand By
01E1  B4 F6                 +               MOV     AH,GTSB
01E3  E8 0009 R             +               CALL    CMD8292         ;GoToStandBy

                                            ;use DMA channel 3
                                            OUTB    15,00AH         ;disable IBM-DMA channels 1 & 3
01E6  B0 0A                 +               MOV     AL,00AH
01E8  BA 000F               +               MOV     DX,15
01EB  EE                    +               OUT     DX,AL
                                            OUTB    PRTIDA,DMACM    ;connect DMA on tecmar board.
01EC  B0 80                 +               MOV     AL,DMACM
01EE  BA 033C               +               MOV     DX,PRTIDA
01F1  EE                    +               OUT     DX,AL
                                            OUTB    INT2,010H       ;enable 8191A for DMA
01F2  B0 10                 +               MOV     AL,010H
01F4  BA 0332               +               MOV     DX,INT2
01F7  EE                    +               OUT     DX,AL
                                            OUTB    AUXMD,PON
01F8  B0 00                 +               MOV     AL,PON
01FA  BA 0335               +               MOV     DX,AUXMD
01FD  EE                    +               OUT     DX,AL

01FE  B0 00                                 MOV     AL,0            ;set DMA Xfer for mem write
0200  33 D2                                 XOR     DX,DX           ; less than 64Kbyte Xfer
0202  E8 0000 E                             CALL    DMA$XFER
                                            OUTB    PRTIDA,0        ;disconnect DMA on tecmar board.
0205  B0 00                 +               MOV     AL,0
0207  BA 033C               +               MOV     DX,PRTIDA
020A  EE                    +               OUT     DX,AL

                                            T_CONTROL_SYN           ;488 bus to control mode
020B  B4 FD                 +               MOV     AH,TCSY
020D  E8 0009 R             +               CALL    CMD8292
0210  B0 80                 +               MOV     AL,AXRA
0212  BA 0335               +               MOV     DX,AUXMD
0215  EE                    +               OUT     DX,AL
0216  B0 03                 +               MOV     AL,FNHSK
0218  BA 0335               +               MOV     DX,AUXMD
021B  EE                    +               OUT     DX,AL
021C  B0 80                 +               MOV     AL,TON
021E  BA 0334               +               MOV     DX,ADRMD
0221  EE                    +               OUT     DX,AL
0222  B0 00                 +               MOV     AL,PON
0224  BA 0335               +               MOV     DX,AUXMD
0227  EE                    +               OUT     DX,AL
```

```
0228  26: C6 04 01                          MOV     BYTE PTR ES:[SI+0],1           ;string size
022C  26: C6 44 01 09                       MOV     BYTE PTR ES:[SI+1],CAMAC_FN+ StartDigt
0231  E8 0026 R                             CALL    CAMACMND                 ;execute camac cycle

0234  9D                                    POPF            ;re_enable interupt requests
0235  1F                                    POP     DS
0236  8B E5                                 MOV     SP,BP
0238  5D                                    POP     BP
0239  C2 000A                               RET     10
023C                        Get1Block       ENDP


023C                        CSEG    ENDS
                                    END     START
```

```
Macros:

                Name                Length

CLR. . . . . . . . . . . . . . .    0001
LISTEN . . . . . . . . . . . . .    0007
LISTEN_FST . . . . . . . . . . .    0008
OUTB . . . . . . . . . . . . . .    0002
OUT_GTSB . . . . . . . . . . . .    0002
OUT_MLA. . . . . . . . . . . . .    0006
OUT_MTA. . . . . . . . . . . . .    0003
OUT_TCSY . . . . . . . . . . . .    0006
OUT_TCSY_FST . . . . . . . . . .    0006
READ_AL. . . . . . . . . . . . .    0003
SETF . . . . . . . . . . . . . .    0001
TALK . . . . . . . . . . . . . .    0005
T_CONTROL_SYN. . . . . . . . . .    0002
UNLISTEN . . . . . . . . . . . .    0003
WAITI. . . . . . . . . . . . . .    0004
WAITO. . . . . . . . . . . . . .    0004
WAITT. . . . . . . . . . . . . .    0003
WAIT_COM . . . . . . . . . . . .    0004
WRITE_AL . . . . . . . . . . . .    0003

Segments and groups:

                Name                Size   align   combine class

CSEG . . . . . . . . . . . . . .    023C   PARA    PUBLIC

Symbols:

                Name                Type   Value   Attr

ABORT. . . . . . . . . . . . . .    Number 00F9
ADR01. . . . . . . . . . . . . .    Number 0336
ADRMD. . . . . . . . . . . . . .    Number 0334
ADRST. . . . . . . . . . . . . .    Number 0334
AUXMD. . . . . . . . . . . . . .    Number 0335
AXRA . . . . . . . . . . . . . .    Number 0000
AXRB . . . . . . . . . . . . . .    Number 00A0
BEGIFTHEN. . . . . . . . . . . .    L NEAR 0040    CSEG
BIM. . . . . . . . . . . . . . .    Number 0001
BOICM. . . . . . . . . . . . . .    Number 0010
BOM. . . . . . . . . . . . . . .    Number 0002
BUSST. . . . . . . . . . . . . .    Number 0338
BYTERET. . . . . . . . . . . . .    Number 0060
CAMCY. . . . . . . . . . . . . .    Number 0003
CAMACMND . . . . . . . . . . . .    N PROC 0026    CSEG     Length =0079
CAMACOM. . . . . . . . . . . . .    N PROC 017C    CSEG     Length =000F
CAMAC_FLAGS. . . . . . . . . . .    Number 0020
CAMAC_FN . . . . . . . . . . . .    Number 0000
CFLAGMASK. . . . . . . . . . . .    Number 0002
CLKRT. . . . . . . . . . . . . .    Number 0025
CLRLAM . . . . . . . . . . . . .    Number 000A
```

```
CLRST. . . . . . . . . . . . . .        Number  0336
CMD8292. . . . . . . . . . . . .        N PROC  0009    CSEG    Length =0010
CMD92. . . . . . . . . . . . . .        Number  0339
CPT. . . . . . . . . . . . . . .        Number  0060
CPTEN. . . . . . . . . . . . . .        Number  0001
CPTRG. . . . . . . . . . . . . .        Number  0335
CRATE_ADDR . . . . . . . . . . .        Number  000A
DEFLT. . . . . . . . . . . . . .        Number  0000
DIN. . . . . . . . . . . . . . .        Number  0330
DMA$XFER . . . . . . . . . . . .        L NEAR  0000    CSEG    External
DMACM. . . . . . . . . . . . . .        Number  0080
DOUT . . . . . . . . . . . . . .        Number  0330
DSBLLAM. . . . . . . . . . . . .        Number  0018
DTDL1. . . . . . . . . . . . . .        Number  0060
DTDL2. . . . . . . . . . . . . .        Number  00E0
EDEOS. . . . . . . . . . . . . .        Number  0004
ENBLDISP . . . . . . . . . . . .        Number  0012
ENBLLAM. . . . . . . . . . . . .        Number  001A
ENBLREAD . . . . . . . . . . . .        Number  0011
ENDIFTHEN. . . . . . . . . . . .        L NEAR  008C    CSEG
ENDMK. . . . . . . . . . . . . .        Number  0010
EOIS . . . . . . . . . . . . . .        Number  0008
EOIST. . . . . . . . . . . . . .        Number  0020
EOSR . . . . . . . . . . . . . .        Number  0337
ERFLG. . . . . . . . . . . . . .        Number  0338
ERRBIT . . . . . . . . . . . . .        Number  0040
ERRM . . . . . . . . . . . . . .        Number  0339
EVBIT. . . . . . . . . . . . . .        Number  0010
EVREG. . . . . . . . . . . . . .        Number  0338
EXPF . . . . . . . . . . . . . .        Number  00F5
FNHSK. . . . . . . . . . . . . .        Number  0003
GET. . . . . . . . . . . . . . .        Number  0008
GET1BLOCK. . . . . . . . . . . .        N PROC  018B    CSEG    Length =00B1
GIDL . . . . . . . . . . . . . .        Number  00F1
GSEC . . . . . . . . . . . . . .        Number  00F4
GTSB . . . . . . . . . . . . . .        Number  00F6
HISPEED. . . . . . . . . . . . .        Number  0008
HLDFOR8037 . . . . . . . . . . .        N PROC  009F    CSEG    Length =0086
HOEND. . . . . . . . . . . . . .        Number  0002
HOLDHERE . . . . . . . . . . . .        L NEAR  00C1    CSEG
HOSHK. . . . . . . . . . . . . .        Number  0001
IACK . . . . . . . . . . . . . .        Number  000B
IBFBT. . . . . . . . . . . . . .        Number  0002
IBFICM . . . . . . . . . . . . .        Number  0008
IFCRBIT. . . . . . . . . . . . .        Number  0004
IFLAGMASK. . . . . . . . . . . .        Number  0008
INIT . . . . . . . . . . . . . .        N PROC  0125    CSEG    Length =0057
INT1 . . . . . . . . . . . . . .        Number  0331
INT2 . . . . . . . . . . . . . .        Number  0332
INTM . . . . . . . . . . . . . .        Number  00A0
INTM1. . . . . . . . . . . . . .        Number  0331
INTMR. . . . . . . . . . . . . .        Number  0338
INTST. . . . . . . . . . . . . .        Number  0339
LA . . . . . . . . . . . . . . .        Number  0001
LAMENBL. . . . . . . . . . . . .        Number  0001
```

```
LON. . . . . . . . . . . . . . .        Number  0040
DA. . . . . . . . . . . . . . .         Number  0001
LA. . . . . . . . . . . . . . .         Number  0021
MODE1. . . . . . . . . . . . . .        Number  0001
TA. . . . . . . . . . . . . . .         Number  0041
DOENBL. . . . . . . . . . . . .         Number  0002
NOXENBL. . . . . . . . . . . . .        Number  0004
VCMD. . . . . . . . . . . . . .         Number  0007
BFBIT . . . . . . . . . . . . .         Number  0001
DBFICM . . . . . . . . . . . . .        Number  0004
UTDATA. . . . . . . . . . . . .         L NEAR  0056    CSEG
ATCHPAUSE . . . . . . . . . . .         L NEAR  0175    CSEG
ON. . . . . . . . . . . . . . .         Number  0000
FPC. . . . . . . . . . . . . . .        Number  0005
PD. . . . . . . . . . . . . . .         Number  0070
PE. . . . . . . . . . . . . . .         Number  0060
PPU. . . . . . . . . . . . . . .        Number  0015
RT91. . . . . . . . . . . . . .         Number  0330
RT92. . . . . . . . . . . . . .         Number  0338
PRTIDA . . . . . . . . . . . . .        Number  033C
DFLAGMASK. . . . . . . . . . . .        Number  0002
BST . . . . . . . . . . . . . .         Number  00E7
CST . . . . . . . . . . . . . .         Number  00E6
RERF . . . . . . . . . . . . . .        Number  00E4
ERM . . . . . . . . . . . . . .         Number  00EA
ESET91. . . . . . . . . . . . .         Number  0002
RET2BYTE . . . . . . . . . . . .        Number  0001
RET3BYTE . . . . . . . . . . . .        Number  0002
ET4BYTE . . . . . . . . . . . .         Number  00A4
REVC . . . . . . . . . . . . . .        Number  00E3
RINM . . . . . . . . . . . . . .        Number  00E5
MEMDATA . . . . . . . . . . . .         Number  0002
MODULE. . . . . . . . . . . . .         Number  0003
RPARMS . . . . . . . . . . . . .        Number  0000
SAMPLE. . . . . . . . . . . . .         Number  0001
SET . . . . . . . . . . . . . .         Number  00F2
RSTI . . . . . . . . . . . . . .        Number  00F3
RTOUT. . . . . . . . . . . . . .        Number  00E9
DC. . . . . . . . . . . . . . .         Number  0004
SDEOI. . . . . . . . . . . . . .        Number  0006
SLOC . . . . . . . . . . . . . .        Number  00F7
PCMI. . . . . . . . . . . . . .         Number  00F0
PD. . . . . . . . . . . . . . .         Number  0019
SPE. . . . . . . . . . . . . . .        Number  0018
PICM. . . . . . . . . . . . . .         Number  0020
REM . . . . . . . . . . . . . .         Number  00F8
SRQBT. . . . . . . . . . . . . .        Number  0020
SRQCNTRL . . . . . . . . . . . .        Number  0040
SDIGTIZE. . . . . . . . . . . .         Number  0008
START. . . . . . . . . . . . . .        L NEAR  0000    CSEG
STARTDIGT. . . . . . . . . . . .        Number  0009
TCMI. . . . . . . . . . . . . .         Number  00FE
TOPDIGT . . . . . . . . . . . .         Number  0019
SYCBIT . . . . . . . . . . . . .        Number  0080
TA . . . . . . . . . . . . . . .        Number  0002
```

```
TCASY. . . . . . . . . . . . . . . .      Number  00FC
TCICM. . . . . . . . . . . . . . . .      Number  0002
TCILC. . . . . . . . . . . . . . . .      Number  033A
TCIN . . . . . . . . . . . . . . . .      Number  0001
TCNTR. . . . . . . . . . . . . . . .      Number  00FA
TCSY . . . . . . . . . . . . . . . .      Number  00FD
TCT. . . . . . . . . . . . . . . . .      Number  0009
TON. . . . . . . . . . . . . . . . .      Number  0080
TOREG. . . . . . . . . . . . . . . .      Number  0338
TOUT1. . . . . . . . . . . . . . . .      Number  0001
TOUT2. . . . . . . . . . . . . . . .      Number  0002
TOUT3. . . . . . . . . . . . . . . .      Number  0004
TR6837SLOT . . . . . . . . . . . . .      Number  0006
TSTLAM . . . . . . . . . . . . . . .      Number  0012
TSTMLAM. . . . . . . . . . . . . . .      Number  0008
UNL. . . . . . . . . . . . . . . . .      Number  003F
VSCMD. . . . . . . . . . . . . . . .      Number  000F
WEVC . . . . . . . . . . . . . . . .      Number  00E2
WOUT . . . . . . . . . . . . . . . .      Number  00E1
WPARMS . . . . . . . . . . . . . . .      Number  0010
XFLAGMASK. . . . . . . . . . . . . .      Number  0001
ZFLAGMASK. . . . . . . . . . . . . .      Number  0001
??0000 . . . . . . . . . . . . . . .      L NEAR  000C    CSEG
??0001 . . . . . . . . . . . . . . .      L NEAR  0020    CSEG
??0002 . . . . . . . . . . . . . . .      L NEAR  002B    CSEG
??0003 . . . . . . . . . . . . . . .      L NEAR  0043    CSEG
??0004 . . . . . . . . . . . . . . .      L NEAR  0051    CSEG
??0005 . . . . . . . . . . . . . . .      L NEAR  0066    CSEG
??0006 . . . . . . . . . . . . . . .      L NEAR  008F    CSEG
??0007 . . . . . . . . . . . . . . .      L NEAR  00C7    CSEG
??0008 . . . . . . . . . . . . . . .      L NEAR  00EC    CSEG
??0009 . . . . . . . . . . . . . . .      L NEAR  00FB    CSEG
??000A . . . . . . . . . . . . . . .      L NEAR  012B    CSEG
??000B . . . . . . . . . . . . . . .      L NEAR  013F    CSEG
??000C . . . . . . . . . . . . . . .      L NEAR  01C4    CSEG

Warning Severe
Errors  Errors
0       0
```

```
PAGE 60,132
```

```
0000                       CSEG     SEGMENT PARA PUBLIC
                                    ASSUME CS:CSEG
0100                                ORG    0100H
                           ;-------------------------------
                           ;        E Q U A T E S         :
                           ;-------------------------------
= 007F                     DMAPAGE  EQU    07FH            ;Base I/O Address of DMA page latch (NO 0)
= 0000                     DMA      EQU    0               ;Base I/O Address of 8237 DMA chip
= 0003                     CHNL     EQU    3               ;DMA CHANNEL NUMBER (CONFIGURABLE, 1..3)
= 0006                     CHAN     EQU    CHNL + 2        ;    I/O Offset for DMA channel
= 0000                     ADDR     EQU    0               ;    I/O channel select, DMA registers
= 0001                     COUNT    EQU    1               ;     "    "     "     "     "
                                                           ;    Offsets for DMA control registers
= 0008                     COMMAND  EQU    8               ;    (W/O)
=                          STATUS   EQU    COMMAND         ;    (R/O)
= 0009                     DRQ      EQU    9               ;    (W/O)
= 000A                     MASKBIT  EQU    10              ;    (W/O)
= 000B                     MODE     EQU    11              ;    (W/O)
= 000C                     FLIPFLOP EQU    12              ;    (W/O)
= 000E                     CLRMASK  EQU    14              ;    (W/O)
= 000F                     MASK     EQU    15              ;    (W/O)

0100                       START:
                           ;-----------------------------------------------------------------
                           ; DMA TRANSFER                                                    :
                           ;      THIS ROUTINE PROVIDES A MEANS OF USING DMA TO TRANSFER A   :
                           ;      PROGRAMMED NUMBER OF BYTES TO OR FROM AN I/O PORT. THE PORT :
                           ;      MUST HAVE THE REQUIRED HARDWARE SUPPORT FOR DMA ACCESS.    :
                           ;      ANY PENDING DMA PROCESS ON THE DMA CHANNEL SPECIFIED BY THE :
                           ;      "CHNL" CONSTANT WILL BE ABORTED. DO NOT USE WITHIN AN      :
                           ;      INTERRUPT HANDLER. ( channel 2 used by 5 1/4" diskettes,   :
                           ;      3 by hard disk)                                            :
                           ; INPUT                                                           :
                           ;      DS:DI - Seg:Ofs pointer to Xfer Address.                   :
                           ;      DX,BX - Number of bytes to Xfer.                           :
                           ;      AL    - DMA Mode byte( 0 = Write mem, 1= Read )            :
                           ; OUTPUT                                                          :
                           ;      CY = 0  SUCCESS                                            :
                           ;      CY = 1  FAILURE                                            :
                           ;      DS:DI - SEG:OFS POINTER TO LAST BYTE TRANSFERRED + 1       :
                           ;      DX,BX - NUMBER BYTES NOT TRANSFERRED                       :
                           ;      (AX)    DESTROYED                                          :
                           ;-----------------------------------------------------------------
0100                       DMA$XFER PROC  NEAR
                                    PUBLIC DMA$XFER
0100  51                            PUSH  CX
0101  8A E0                         MOV   AH,AL
0103  E4 08                         IN    AL,DMA.STATUS   ;Check DMA request FLAG
0105  24 80                         AND   AL,080H         ;(10H SHL CHNL)
0107  75 04                         JNZ   DMAX$00
0109  F9                            STC                   ;NO DMA PROCESS PENDING, FLAG ERROR
010A  E9 0199 R                     JMP   DMAX$END        ;   & ABBORT
010D                       DMAX$00:                       ;PROGRAM DMA MODE
010D  B0 47                         MOV   AL,044H + CHNL  ; - BLOCK SINGLE MEM. WRITE
010F  0A E4                         OR    AH,AH
```

```
0111  74 0A                     JZ    DMAX000
0113  B0 48                     MOV   AL,048H + CHNL  ; - BLOCK SINGLE MEM. READ
0115  FE CC                     DEC   AH
0117  74 04                     JZ    DMAX000
0119  F9                        STC                   ; - UNKNOWN MODE, FLAG ERROR
011A  EB 7D 90                  JMP   DMAX$END        ;   & ABBORT
011D                   DMAX000:
011D  FA                        CLI                   ;no more interupts
011E  E6 0B                     OUT   DMA.MODE,AL
0120  B0 07                     MOV   AL,CHNL + 4     ;DISABLE DMA CHANNEL
0122  E6 0A                     OUT   DMA.MASKBIT,AL
                                ;convert Seg:Ofs address of the data transfer area
                                ;in DS:DI to an absolute address in CH,DI.
0124  8C D8                     MOV   AX,DS           ;get Seg value
0126  B1 04                     MOV   CL,4
0128  D3 C0                     ROL   AX,CL           ;convert to an Ofs.
012A  8A E8                     MOV   CH,AL           ;save highest nybble of Addr.
012C  24 F0                     AND   AL,0F0H
012E  03 F8                     ADD   DI,AX           ;add to Ofs value
0130  73 02                     JNC   DMAX001
0132  FE C5                     INC   CH              ;addition overflow
0134  80 E5 0F         DMAX001: AND   CH,0FH
                                ;Calculate Maximum number bytes DMA can read/write on
                                ;1st 64K page
0137  8B C7                     MOV   AX,DI
0139  F7 D8                     NEG   AX
                                ; check if # bytes to Xfer > Maximum current 64K page
013B  0B D2                     OR    DX,DX
013D  75 0A                     JNZ   XFERAX
013F  0B C0                     OR    AX,AX
0141  74 04                     JZ    DMAX002
0143  3B D8                     CMP   BX,AX
0145  73 02                     JAE   XFERAX
0147                   DMAX002: ;Set # bytes to xfer = Xfer count (DX=0)
0147  8B C3                     MOV   AX,BX
0149                   XFERAX:  ;The following code is used to Xfer AX # bytes using
                                ;DMA to/from the memory pointed to by the absolute
                                ;address contained in DS:DI.
0149  90                        NOP
014A  90                        NOP
014B  50                        PUSH  AX              ;SAVE CURRENT COUNT
014C  48                        DEC   AX              ;COUNT-1 for DMA register
014D  E6 0C                     OUT   DMA.FLIPFLOP,AL ;reset low/high flip flop.
014F  E6 07                     OUT   DMA.CHAN.COUNT,AL ;LOW BYTE OF COUNT
0151  8A C4                     MOV   AL,AH
0153  E6 07                     OUT   DMA.CHAN.COUNT,AL ;HIGH BYTE OF COUNT
0155  8B C7                     MOV   AX,DI
0157  E6 06                     OUT   DMA.CHAN.ADDR,AL  ;LOW BYTE OF ADDRESS
0159  8A C4                     MOV   AL,AH
015B  E6 06                     OUT   DMA.CHAN.ADDR,AL  ;HIGH BYTE OF ADDRESS
015D  8A C5                     MOV   AL,CH           ;GET HIGH 4 BITS ADDRESS
015F  E6 82                     OUT   DMAPAGE.CHNL,AL ;PROGRAM PAGE REGISTER
0161  B0 03                     MOV   AL,CHNL         ;ENABLE CHANNEL FOR DMA
0163  E6 0A                     OUT   DMA.MASKBIT,AL
0165                   HLDTC:
```

```
0165  E4 08                        IN    AL,DMA.STATUS   ;WAIT FOR TC FLAG
0167  24 08                        AND   AL,8            ;1 SHL CHNL
0169  74 FA                        JZ    HLDTC

016B  58                           POP   AX
016C  90                           NOP
016D  90                           NOP
                                   ;update absolute memory address & number bytes to xfer.
016E  0B C0                        OR    AX,AX
0170  75 06                        JNZ   PARTPAGE
0172  FE C5                        INC   CH
0174  4A                           DEC   DX
0175  EB 0C 90                     JMP   FINIUP
0178  03 F8               PARTPAGE:ADD   DI,AX
017A  73 02                        JNC   NOOV
017C  FE C5                        INC   CH
017E  2B D8               NOOV:    SUB   BX,AX
0180  73 01                        JNC   FINIUP
0182  4A                           DEC   DX
0183                      FINIUP:
0183  33 C0                        XOR   AX,AX           ;Default 64Kbyte Xfer
0185  0B D2                        OR    DX,DX
0187  75 C0                        JNZ   XFERAX          ;more than 64Kbytes to go...
0189  0B DB                        OR    BX,BX
018B  75 BA                        JNZ   DMAX002         ;BX bytes to go

018D  FB                           STI                   ;ENABLE INTERRUPTS AGAIN
018E  B1 04                        MOV   CL,4            ;CONVERT HIGH 4 BITS ADDRESS
0190  D2 CD                        ROR   CH,CL           ;   TO SEGMENT FORMAT
0192  81 E1 F000                   AND   CX,0F000H
0196  8E D9                        MOV   DS,CX           ;RESULTS TO SEGMENT POINTER
0198  F8                           CLC                   ;FLAG DMA XFER SUCCESSFUL
0199                      DMAX$END:
0199  59                           POP CX
019A  C3                           RET

019B                      DMA$XFER ENDP
019B                      CSEG     ENDS
                                   END   START
```

```
The IBM Personal Computer MACRO Assembler 02-02-87      PAGE    Symbols-1


Segments and groups:

                N a m e                 Size    align   combine class

CSEG . . . . . . . . . . . . . .        019B    PARA    PUBLIC

Symbols:

                N a m e                 Type    Value   Attr

ADDR . . . . . . . . . . . . . .        Number  0000
CHAN . . . . . . . . . . . . . .        Number  0006
CHNL . . . . . . . . . . . . . .        Number  0003
CLRMASK. . . . . . . . . . . . .        Number  000E
COMMAND. . . . . . . . . . . . .        Number  0008
COUNT. . . . . . . . . . . . . .        Number  0001
DMA. . . . . . . . . . . . . . .        Number  0000
DMA$XFER . . . . . . . . . . . .        N PROC  0100    CSEG    Global  Length =009B
DMAPAGE. . . . . . . . . . . . .        Number  007F
DMAX$00. . . . . . . . . . . . .        L NEAR  010D    CSEG
DMAX$END . . . . . . . . . . . .        L NEAR  0199    CSEG
DMAX000. . . . . . . . . . . . .        L NEAR  011D    CSEG
DMAX001. . . . . . . . . . . . .        L NEAR  0134    CSEG
DMAX002. . . . . . . . . . . . .        L NEAR  0147    CSEG
DRQ. . . . . . . . . . . . . . .        Number  0009
FINIUP . . . . . . . . . . . . .        L NEAR  0183    CSEG
FLIPFLOP . . . . . . . . . . . .        Number  000C
HLDTC. . . . . . . . . . . . . .        L NEAR  0165    CSEG
MASK . . . . . . . . . . . . . .        Number  000F
MASKBIT. . . . . . . . . . . . .        Number  000A
MODE . . . . . . . . . . . . . .        Number  000B
NOOV . . . . . . . . . . . . . .        L NEAR  017E    CSEG
PARTPAGE . . . . . . . . . . . .        L NEAR  0178    CSEG
START. . . . . . . . . . . . . .        L NEAR  0100    CSEG
STATUS . . . . . . . . . . . . .        Alias   COMMAND
XFERAX . . . . . . . . . . . . .        L NEAR  0149    CSEG

Warning Severe
Errors  Errors
0       0
```

```
PROGRAM NDT_DATA(CONSOLE,GBIP,AUXPORT);
{$R-}
CONST
   RevisionDate        = 'January 8, 1987';
   DefaultDataDrivePath= 'A:';
   NumChannels         = 80;
   FCTN = 1;   CHANNEL = 2;   SLOT = 3;
   DATAL= 4;   DataM   = 5;   DataH= 6;
   MaxRecordRate       = 100E6; {MHz, sampling rate}
   BlockSize           = 2;    {Kbyte digitizer memory allocation unit}
   ClkRate             = 6E7;  {MHz, phase system clock rate}
   TR_Slot             = 6;
   AUXPORT             = $300;    {IBM PROTOTYPE CARD ($300..$30F)}
   { . . . . . . . A U X P O R T . . . . . . . }
   {         + D0..D3 = || Phase ||             }
   {    Tx  -| D4,D5  = Register Select         }
   {         | D6     = Direction bit           }
   {         + D7     = Enable                  }
   {                                            }
   {    Rx  -+ D0..D6 = Address Bits (0..79)    }
   {         + D7     = -Enable                 }
   { . . . . . . . . . . . . . . . . . . . . . }

TYPE
   CmndString         = STRING[6];
   FileNameTYPE       = String[13];
   StringTYPE         = STRING[255];
   DataBlock          = ARRAY[1..$1000] OF BYTE;
   DataBlockPNTR      = ^DataBlock;
   DatumTYPE          = ARRAY[1..NumChannels] OF DataBlockPntr;
   ChannelSet         = SET OF 1..NumChannels;
   DigitizerCodeTYPE = INTEGER; {word}
   RegDesigTYPE       = (A,B);

CONST (TYPED)
   AllChannels   : ChannelSet = [1..NumChannels];

VAR
   CH: CHAR;
   Theta      : INTEGER; { Requested Beam Angles }
   Phi        : REAL;
   DELAYCNT   : BYTE;
   VELOCITY   : REAL;
   Spacing    : REAL; { Meters }
   Adir,Bdir  : -1..1;
   I,J,K      : INTEGER;
   Dummy      : CmndString;
   Command    : CmndString;
   Datum      : DatumTYPE ;
   MemError   : BOOLEAN;
   DATA       : FILE OF DataBlock;
   HEADER     : TEXT;

PROCEDURE GPIBCAMAC;                                 External 'GPICAMAC.BIN';
PROCEDURE InitGPIB;                                  External GPIBCAMAC[0];
PROCEDURE Exec      ( VAR Cmnd : CmndString);   External GPIBCAMAC[3];
PROCEDURE ReadBlock( VAR ARY  : DataBlock;
                        NumBytes : INTEGER;
                        VAR Dummy: CondString {: External GPIBCAMAC[6];

PROCEDURE LOAD(Desig:RegDesigTYPE;Val:BYTE);
   { - This procedure corrects a bug with the hardware in the direction
   bit logic. The software solution requires the phase count to be altered..}
   BEGIN
   CASE Desig OF
      A: BEGIN
         Val:= (Val AND $40)+(((Val XOR $F) +1) AND $F);
         {               en    sel   Dir&phase }
         port[AuxPort  ]:= $80 + $10 + Val;
         port[AuxPort+1]:= 0; {LDFR}
         port[AuxPort+1]:= 0;
         END;
      B: BEGIN
         Val:= (Val AND $40)+(((Val XOR $F) +1) AND $F);
         {               en    sel   Dir/phase }
         port[AuxPort  ]:= $80 + $20 + Val;
         port[AuxPort+1]:= 0; {LDFR}
         port[AuxPort+1]:= 0;
         END;
      END;{case}
   END;{PROC. LOAD}

PROCEDURE ERROR(ErrNo: INTEGER);
   BEGIN
   IF ErrNo <> 0 THEN
      WRITELN('ERROR # ',ErrNo);
   {HALT;}
   END;{PROC. Error}

PROCEDURE OperationMessage( MsgString : StringType);
   BEGIN
   GoToXY(1,8);
   WRITE(MsgString);
   END;{PROC. OperationMessage}

PROCEDURE SystemCheck;
   BEGIN
   PORT[AUXPORT]:= 0;
   IF PORT[AUXPORT] <> 0 THEN ERROR(2);
   PORT[AUXPORT]:= $FF;
   IF PORT[AUXPORT] <> $FF THEN ERROR(3);
   END;{PROC. SystemCheck}

PROCEDURE ASSIGN_Datum(VAR DATUM:DatumTYPE; VAR OUTOFMEM:BOOLEAN);
{  This procedure is used for assigning pointers to 4K Bytes blocks that
do NOT have an absolute 64K page break contained in them.  This condition
is essential as DMA cannot read into a block  extending accross a 64K page
break, and maximum block size for this program is 8K Bytes. }

VAR
   VALID:BOOLEAN;
   I,J:BYTE;
   Unused: ARRAY[1..100] OF DataBlockPntr;

```

```

PROCEDURE ON64KPAGE(VAR PNTR:DataBlockPNTR; VAR VALID:BOOLEAN);
VAR
   SEGMENT,OFFSET: INTEGER;
   I: BYTE;
   OLDpntrVALUE,NEWpntrVALUE: real;
BEGIN
OLDpntrVALUE:= SEG(PNTR^)*16 + OFS(PNTR^);
SEGMENT:= (SEG(PNTR^) AND $F000);
OFFSET:= OFS(PNTR^) + ((SEG(PNTR^) AND $0FFF) SHL 4);
IF (OFFSET<0) AND (OFFSET+SizeOf(Pntr^)>=0) THEN {BLOCK extends past 64K boundry}
   VALID:= FALSE
ELSE
   BEGIN
   VALID:=TRUE;
   PNTR:= PTR(SEGMENT,OFFSET)
   END;
NEWpntrVALUE:= SEG(PNTR^)*16 + OFS(PNTR^);
IF NEWpntrVALUE <> OLDpntrVALUE THEN
   BEGIN
   WRITELN('PONTER RE-ASSIGNMENT ERROR',^G);
   HALT;
   END;
{PROC. ON64KPAGEEND};
begin {---ASSIGN---}
OUTOFMEM:= FALSE;
J:=0;
FOR I:= 1 TO NumChannels DO REPEAT
   IF (MaxAvail>= SizeOf(DATUM[1]^) SHR 4) OR (MaxAvail < 0) THEN
      BEGIN
      NEW(DATUM[I]);
      ON64KPAGE(DATUM[I],VALID);
      IF Not VALID THEN
         BEGIN
         J:= J + 1;
         Unused[J]:= Datum[I];
         END;
      END
   ELSE
      OUTOFMEM:=TRUE;
   UNTIL VALID OR OUTOFMEM;
FOR I:= 1 TO J DO
   Dispose(Unused[I]);
{Proc. ASSIGNend};

PROCEDURE DisplayTitle;
   VAR DisplayTimeCount: INTEGER;
   BEGIN
   CLRSCR;
   GoToXY(1,11);
   WRITELN('                          NDT Phased Array, Data Acquistion Subsystem.');
   WRITELN('                              Version: Alpha test':50);
   GoToXY(1,22);
   WRITELN(' Programed for Dr. H.W. Jones by T.E. Hayman,');
   WRITELN(' Arctec Canada Ltd Subcontract.',RevisionDate:43);
   DisplayTimeCount:= 100;
   REPEAT
```

```
      DELAY(100);
      DisplayTimeCount:= DisplayTimeCount - 1;
      UNTIL KeyPressed OR (DisplayTimeCount=0);
   END;{PROC. DisplayTitle}

PROCEDURE SetOptions;
   BEGIN
   ClrScr;
   WRITELN('IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM:
   WRITELN(':                                                                        :
   WRITELN(':                         O P T I O N S   M E N U                        :
   WRITELN(':                                                                        :
   WRITELN('LMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
   WRITELN(':                                                                        :
   WRITELN(':        V : Velocity Select                                             :
   WRITELN(':        Q : Quit menu                                                   :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN(':                                                                        :
   WRITELN('GDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD:
   WRITELN(':  SELECTION:                                                            :
   WRITE  ('HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
   REPEAT
```

```
      GOTOXY(15,23);
      WRITE(' ':50);
      GOTOXY(15,23);
      READ(KBD,CH);
      CH:= UpCase(CH);
      CASE CH OF
         'V': BEGIN
                WRITE(' VELOCITY [M/S] : ');
                READ(VELOCITY)
                END
         END;{case}
      UNTIL CH = 'Q';
   END;{PROC. SetOptions}

PROCEDURE PromptUser;
   BEGIN
   CLRSCR;
   GoToXY(1,6);
   WRITELN('                              Data Acquisition System');
   GoToXY(1,22);
   WRITELN(' i = Beam Angle from defined direction on array surface');
   WRITELN(' ■ = Beam Angle from normal to array surface');
   REPEAT
      GoToXY(1,20);
      WRITE('Tx direction [i,■]: '); ClrEol;
      {$I-} READLN(Theta,Phi);  {$I+}
      UNTIL IOresult = 0;

  {Convert angles to standard form}
   Theta:= Theta MOD 360;
   Phi:= Frac(Phi/360) * 360;

   WHILE (ABS(Theta)<>0  ) AND (ABS(Theta)<>45 ) AND (ABS(Theta)<> 90) AND
         (ABS(Theta)<>135) AND (ABS(Theta)<>180) AND (ABS(Theta)<>225) AND
         (ABS(Theta)<>270) AND (ABS(Theta)<>315) DO
      REPEAT
         GoToXY(1,20);
         WRITE('Tx direction [i] = [0,45,90,135,180,225,270,315] : '); ClrEol;
         {$I-} READLN(Theta); {$I+}
         Theta:= Theta MOD 360;
         UNTIL IOresult = 0;
   IF Phi < 0 THEN
      BEGIN
      Phi:= -Phi;
      Theta:= (Theta + 180) MOD 360;
      END;
   END;{PROC. PromptUser}

PROCEDURE SetControls;
   BEGIN
   IF Theta=0 THEN
      BEGIN
      Spacing:= 2E-3;
      Adir:=+1;
      Bdir:=+1;
      END
   ELSE IF Theta= 45 THEN
```

```
      BEGIN
      Spacing:= 2E-3 * SQRT(2);
      Adir:=+1;
      Bdir:= 0;
      END
   ELSE IF Theta= 90 THEN
      BEGIN
      Spacing:= 2E-3;
      Adir:=+1;
      Bdir:=-1;
      END
   ELSE IF Theta=135 THEN
      BEGIN
      Spacing:= 2E-3 * SQRT(2);
      Adir:= 0;
      Bdir:=-1;
      END
   ELSE IF Theta=180 THEN
      BEGIN
      Spacing:= 2E-3;
      Adir:=-1;
      Bdir:=-1;
      END
   ELSE IF Theta=225 THEN
      BEGIN
      Spacing:= 2E-3 * SQRT(2);
      Adir:=-1;
      Bdir:= 0;
      END
   ELSE IF Theta=270 THEN
      BEGIN
      Spacing:= 2E-3;
      Adir:=-1;
      Bdir:=+1;
      END
   ELSE  { Theta=315 }
      BEGIN
      Spacing:= 2E-3 * SQRT(2);
      Adir:= 0;
      Bdir:=+1;
      END
   END;{PROC. SetControls}

FUNCTION ARCSIN(SV: REAL): REAL;
   BEGIN
   IF SV = 0 THEN ARCSIN:= 0
   ELSE IF SV=+1 THEN ARCSIN:= +PI/2
   ELSE IF SV=-1 THEN ARCSIN:= -PI/2
   ELSE ARCSIN:= ARCTAN(SV/SQRT(1-SQR(SV)));
   END; {FUNC. ARCSIN}

FUNCTION Encode( RecordRate: REAL; {MHz}
                 MemorySize: BYTE; {KBytes}
                 PreTrigger: BYTE  {1/8'ths} ): DigitizerCodeTYPE;
   VAR
      Code : DigitizerCodeTYPE;
   BEGIN
   IF RecordRate > MaxRecordRate THEN
      BEGIN
      WRITELN;
      WRITELN('ERROR ==> Digitization rate exceeds device capabilities');
      WRITELN('program terminated');
      HALT;
      END
   ELSE IF RecordRate = MaxRecordRate THEN
      code:= 0
   ELSE IF RecordRate = MaxRecordRate/2 THEN
      code:= $10
   ELSE IF RecordRate = MaxRecordRate/4 THEN
      code:= $20
   ELSE IF RecordRate = MaxRecordRate/8 THEN
      code:= $30
   ELSE IF RecordRate = MaxRecordRate/16 THEN
      code:= $40
   ELSE IF RecordRate = MaxRecordRate/32 THEN
      code:= $50
   ELSE IF RecordRate = MaxRecordRate/64 THEN
      code:= $60
   ELSE
      code:= $70; {external clock code}
   IF MemorySize <> 0 THEN
      MemorySize:= ((MemorySize-1) DIV BlockSize) AND $3F;
   PreTrigger:= PreTrigger AND $0F;
   Encode:= (MemorySize SHL 8) + Code + PreTrigger;
   END;{FUNC. Encode}

FUNCTION FileName(Theta,Phi: REAL): FileNameTYPE;
   VAR
      Str1,Str2: STRING[4];
      Code     : INTEGER;
   BEGIN
   STR(Theta:1:0,Str1);
   STR(Phi  :3:1,Str2);
   Str2[POS('.',Str2)]:= '_';
   FileName:= Str1 + '&' + Str2;
   END;{FUNC. FileName}

PROCEDURE AcquireDataSets( RecordSet: ChannelSet );
   VAR
      I,J    : INTEGER;
      Retry  : BYTE;
      DigCode: DigitizerCodeTYPE;
   BEGIN
   OperationMessage('                     Acquiring 80 Sets of data,' + ^M^J
                   +'                     Please Stand By...');

   { Stop digitizer running }
   Command:='123';
   Command[FCTN]:= CHR(25);
   Command[Channel]:= CHR(0);
   Command[Slot]:= CHR(TR_Slot);
   Exec(Command);

   { wait for digitization to finish }
```

```
    DELAY(50);

    { Program digitizer with operational values }
    DigCode:= EnCode(100E6,4,0);
    Command:='12345';
    Command[FCTN]:= CHR(16);
    Command[Channel]:= CHR(0);
    Command[Slot]:= CHR(TR_Slot);
    Command[DataL]:= CHR(LO(DigCode));
    Command[DataM]:= CHR(HI(DigCode));
    Exec(Command);

    { Start digitizer running }
    Command:='123';
    Command[FCTN]:= CHR(9);
    Command[Channel]:= CHR(0);
    Command[Slot]:= CHR(TR_Slot);
    Exec(Command);

    FOR I:= 1 TO NumChannels DO IF I IN RecordSet THEN
        BEGIN
        ReTry:= 0;
        repeat
        {write(i:3,ReTry:3,^H^H^H^H^H^H);}
        { Program Tx Registers }
        IF Adir <> 0 THEN
           LOAD(A,DelayCnt*Adir);
        IF Bdir <> 0 THEN
           LOAD(B,DelayCnt*Bdir);
        { Issue Tx Command      }
        PORT[AuxPort+2]:= 0;          { Strobe signal }
        PORT[AuxPort  ]:= $80 + $20; {this line to be removed on hrdw fix.}
        { Select Analog channel }
        PORT[AuxPort]:= I-1; {WARNING.... there may be bug here,
                                    +16 required if 138 00 line (pin 15) not used.}
        { Read data back into computer }
        ReadBlock(DATUM[I]^,$1000,Dummy);
        ReTry:= ReTry + 1;
        until (datum[I]^[1] <> 0) OR (ReTry = 21);
        if ReTry = 21 then write(^G);
        END;
(*
FOR I:= 1 TO NumChannels DO IF I IN RecordSet THEN
    FOR J:= 1 TO 100 DO
        writeln(I,DATUM[I]^[J]:5);
*)
    OperationMessage(^J + '                              Saving to disk...'); ClrEol;
    { Open data files }
    ASSIGN(DATA,  DefaultDataDrivePath + FileName(Theta,Phi) + '.NDD');
    ASSIGN(HEADER,DefaultDataDrivePath + FileName(Theta,Phi) + '.NDH');
    REWRITE(DATA);
    REWRITE(HEADER);

    { Write Header to disk }
    WRITELN(HEADER,'Theta'   :8, 'Phi'    :10  );
    WRITELN(HEADER, Theta    :8,  Phi     :10:1);
    WRITELN(HEADER,'Velocity':8,'   ', 'ClkRate':8 );
```

```
   WRITELN(HEADER, VELOCITY :8,'    ', ClkRate :8 );
   WRITELN(HEADER,'DATA SET INCLUDES CHANNELS:');
   J:= 0;
   FOR I:= 1 TO NumChannels DO IF I IN RecordSet THEN
      BEGIN
      J:= J+1;
      WRITE(HEADER,I:4);
      IF J = 15 THEN
         BEGIN
         WRITELN(HEADER):
         J:= 0;
         END:
      END;
   CLOSE(HEADER);

   FOR I:= 1 TO NumChannels DO IF I IN RecordSet THEN
      WRITE(DATA,DATUM[I]^);
   CLOSE(DATA);
   OperationMessage(^J + '                    Completed.'); ClrEol;
   END;{PROC. AcquireDataSets}

BEGIN
WRITE('SYSTEM INIT ');
InitGPIB;
{  After power up, bring all camac modules to known state by executing a }
{camac bus cycle with Z and C bus lines set.                             }
Command:= '1';
Command[FCTN]:= CHR($23);
Exec(Command);

SystemCheck;
Assign_Datum(DATUM,MemError);
IF MemError THEN
   BEGIN
   WRITELN('Array allocation error, program terminated.');
   HALT;
   END;
WRITELN('IS NOW COMPLETED');

DELAY(500);
DisplayTitle;
VELOCITY:= 6000;
READ(KBD,CH);
REPEAT
   IF UpCase(CH) = 'O' THEN
      SetOptions;
   PromptUser;
   SetControls;

   { calculate delay count and force into the range 0..15}
   {WARNING... this will require fine tuning to be accurate,
        currently it is assumed that values of 0,1,..,15 corespond
        to delays of 0,10,..150 ns between elements. This may not be the case.}
   DelayCnt:= ROUND(Spacing * Sin(Phi*PI/180)/ Velocity * ClkRate);
   IF DelayCnt > 15 THEN DelayCnt:= 15
   ELSE IF DelayCnt < 0 THEN DelayCnt:= 0;
   { reflect rounded angle to user }
```

```
  PHI:=  180*ARCSIN(DelayCnt/ClkRate * VELOCITY /Spacing)/PI;
  GoToXY(1,20);
  WRITE('Tx direction [i,m] = ',Theta,', ',Phi:3:1); CLREOL;

  AcquireDataSets(AllChannels);

  GoToXY(1,15);
  WRITE  (' Do you wish another set [Y/N] : ');
  Read(KBD,CH);
  SetControls;
  UNTIL UpCase(CH) = 'N';
ClrScr;
writeln('program complete.');

```

```
A . . . . . . . . . . . . . . . . . . .       33      66     402
ABS . . . . . . . . . . . . . . . . . .      237     238     239
ACQUIREDATASETS . . . . . . . . . . . .      359     494
ADIR . . . . . . . . . . . . . . . . .        45     258     264     270     276     282
                             288     294     300     401     402
ALLCHANNELS . . . . . . . . . . . . . .       36     494
ARCSIN . . . . . . . . . . . . . . . .       305     307     308     309     310     490
ARCTAN . . . . . . . . . . . . . . . .       310
ARY . . . . . . . . . . . . . . . . . .       57
ASSIGN . . . . . . . . . . . . . . . .       424     425
ASSIGN_DATUM . . . . . . . . . . . . .       104     464
AUXPORT . . . . . . . . . . . . . . . .        1      13      69      70      71      76
                              77      78      98      99     100     101     406     407
                             409
B . . . . . . . . . . . . . . . . . . .       33      73     404
BDIR . . . . . . . . . . . . . . . . .        45     259     265     271     277     283
                             289     295     301     403     404
BLOCKSIZE . . . . . . . . . . . . . . .       10     343
BOOLEAN . . . . . . . . . . . . . . . .       50     104     111     116
BYTE . . . . . . . . . . . . . . . . .        28      42      61     112     119     314
                             315     362
CH . . . . . . . . . . . . . . . . . .        39     208     209     210     216     475
                             477     498     500
CHANNEL . . . . . . . . . . . . . . . .        7     371     382     391
CHANNELSET . . . . . . . . . . . . . .        31      36     359
CHAR . . . . . . . . . . . . . . . . .        39
CHR . . . . . . . . . . . . . . . . . .      370     371     372     381     382     383
                             384     385     390     391     392     460
CLKRATE . . . . . . . . . . . . . . . .       11     433     486     490
CLOSE . . . . . . . . . . . . . . . . .      446     450
CLREOL . . . . . . . . . . . . . . . .       229     242     422     451     492
CLRSCR . . . . . . . . . . . . . . . .       163     179     221     501
CMND . . . . . . . . . . . . . . . . .        56
CMNDSTRING . . . . . . . . . . . . . .        25      47      48      56      59
CODE . . . . . . . . . . . . . . . . .       317     327     329     331     333     335
                             337     339     341     345     351
COMMAND . . . . . . . . . . . . . . . .       48     369     370     371     372     373
                             380     381     382     383     384     385     386     389
                             390     391     392     393     459     460     461
CONSOLE . . . . . . . . . . . . . . . .        1
DATA . . . . . . . . . . . . . . . . .        51     424     426     449     450
DATABLOCK . . . . . . . . . . . . . . .       28      29      51      57
DATABLOCKPNTR . . . . . . . . . . . . .       29      30     113     116
DATAH . . . . . . . . . . . . . . . . .        8
DATAL . . . . . . . . . . . . . . . . .        8     384
DATAM . . . . . . . . . . . . . . . . .        8     385
DATUM . . . . . . . . . . . . . . . . .       49     104     143     145     146     150
                             412     414     449     464
DATUMTYPE . . . . . . . . . . . . . . .       30      49     104
DEFAULTDATADRIVEPATH . . . . . . . . .         5     424     425
DELAY . . . . . . . . . . . . . . . . .      172     376     472
DELAYCNT . . . . . . . . . . . . . . .        42     402     404     486     487     488
                             490
DESIG . . . . . . . . . . . . . . . . .       61      65
DIGCODE . . . . . . . . . . . . . . . .      363     379     384     385
DIGITIZERCODETYPE . . . . . . . . . . .       32     315     317     363
DISPLAYTIMECOUNT . . . . . . . . . . .       161     170     173     174
```

```
DISPLAYTITLE . . . . . . . . . . . . . . . . . . .         160       473
DISPOSE  . . . . . . . . . . . . . . . . . . . . .         157
DUMMY  . . . . . . . . . . . . . . . . . . . . . .          47        59       412
E  . . . . . . . . . . . . . . . . . . . . . . . .         257       263       269       275       281       287
                                       293       299
E6 . . . . . . . . . . . . . . . . . . . . . . . .           9       379
E7 . . . . . . . . . . . . . . . . . . . . . . . .          11
ENCODE . . . . . . . . . . . . . . . . . . . . . .         313       345       379
ERRNO  . . . . . . . . . . . . . . . . . . . . . .          83        85        86
ERROR  . . . . . . . . . . . . . . . . . . . . . .          83        99       101
EXEC . . . . . . . . . . . . . . . . . . . . . . .          56       373       366       393       461
FALSE  . . . . . . . . . . . . . . . . . . . . . .         126       140
FCTN . . . . . . . . . . . . . . . . . . . . . . .           7       370       381       390       460
FILENAME . . . . . . . . . . . . . . . . . . . . .         348       356       424       425
FILENAMETYPE . . . . . . . . . . . . . . . . . . .          26       348
FRAC . . . . . . . . . . . . . . . . . . . . . . .         235
G  . . . . . . . . . . . . . . . . . . . . . . . .         135       415
GBIP . . . . . . . . . . . . . . . . . . . . . . .           1
GOTOXY . . . . . . . . . . . . . . . . . . . . . .          92       164       167       205       207       222
                                       224       228       241       491       496
GPIBCAMAC  . . . . . . . . . . . . . . . . . . . .          54        55        56        59
HALT . . . . . . . . . . . . . . . . . . . . . . .         136       324       468
HEADER . . . . . . . . . . . . . . . . . . . . . .          52       425       427       430       431       432
                                       433       434       439       442       446
HI . . . . . . . . . . . . . . . . . . . . . . . .         385
I  . . . . . . . . . . . . . . . . . . . . . . . .          46       112       119       142       145       146
                                       150       156       157       361       395       409       412       414
                                       436       439       448       449
INITGPIB . . . . . . . . . . . . . . . . . . . . .          55       456
INTEGER  . . . . . . . . . . . . . . . . . . . . .          32        40        46        58        83       118
                                       161       351       361
IORESULT . . . . . . . . . . . . . . . . . . . . .         231       245
J  . . . . . . . . . . . . . . . . . . . . . . . .          46       112       141       149       150       156
                                       361       365       422       435       438       440       443       451
K  . . . . . . . . . . . . . . . . . . . . . . . .          46
KBD  . . . . . . . . . . . . . . . . . . . . . . .         208       475       498
KEYPRESSED . . . . . . . . . . . . . . . . . . . .         174
LO . . . . . . . . . . . . . . . . . . . . . . . .         384
LOAD . . . . . . . . . . . . . . . . . . . . . . .          61       402       404
M  . . . . . . . . . . . . . . . . . . . . . . . .         365
MAXAVAIL . . . . . . . . . . . . . . . . . . . . .         143
MAXRECORDRATE  . . . . . . . . . . . . . . . . . .           9       319       326       328       330       332
                                       334       336       338
MEMERROR . . . . . . . . . . . . . . . . . . . . .          50       464       465
MEMORYSIZE . . . . . . . . . . . . . . . . . . . .         314       342       343       345
MSGSTRING  . . . . . . . . . . . . . . . . . . . .          90        93
NDT_DATA . . . . . . . . . . . . . . . . . . . . .           1
NEW  . . . . . . . . . . . . . . . . . . . . . . .         145
NEWPNTRVALUE . . . . . . . . . . . . . . . . . . .         120       132       133
NUMBYTES . . . . . . . . . . . . . . . . . . . . .          58
NUMCHANNELS  . . . . . . . . . . . . . . . . . . .           6        30        31        36       142       395
                                       436       448
OFFSET . . . . . . . . . . . . . . . . . . . . . .         118       124       125       130
OFS  . . . . . . . . . . . . . . . . . . . . . . .         122       124       132
OLDPNTRVALUE . . . . . . . . . . . . . . . . . . .         120       122       133
ON64KPAGE  . . . . . . . . . . . . . . . . . . . .         116       146
OPERATIONMESSAGE . . . . . . . . . . . . . . . . .          90       365       422       451
```

```
OUTOFMEM . . . . . . . . . . .    104    140    154    155
PHI . . . . . . . . . . . . .      41    230    235    246    248    348
                      354    424    425    431    486    490    492
PI  . . . . . . . . . . . . .     308    309    486    490
PNTR  . . . . . . . . . . . .     116    122    123    124    125    130
                      132
PORT  . . . . . . . . . . . .      69     70     71     76     77     78
                       98     99    100    101    406    407    409
POS . . . . . . . . . . . . .     355
PRETRIGGER  . . . . . . . . .     315    344    345
PROMPTUSER  . . . . . . . . .     219    479
PTR . . . . . . . . . . . . .     130
READ  . . . . . . . . . . . .     208    213    475    498
READBLOCK . . . . . . . . . .      57    412
READLN  . . . . . . . . . . .     230    243
REAL  . . . . . . . . . . . .      41     43     44    120    305    313
                      348
RECORDRATE  . . . . . . . . .     313    319    326    328    330    332
                      334    336    338
RECORDSET . . . . . . . . . .     359    395    436    448
REGDESIGTYPE  . . . . . . . .      33     61
RETRY . . . . . . . . . . . .     362    397    413    414    415
REVISIONDATE  . . . . . . . .       4    169
REWRITE . . . . . . . . . . .     426    427
ROUND . . . . . . . . . . . .     486
SEG . . . . . . . . . . . . .     122    123    124    132
SEGMENT . . . . . . . . . . .     118    123    130
SETCONTROLS . . . . . . . . .     253    480    499
SETOPTIONS  . . . . . . . . .     177    478
SIN . . . . . . . . . . . . .     486
SIZEOF  . . . . . . . . . . .     125    143
SLOT  . . . . . . . . . . . .       7    372    383    392
SPACING . . . . . . . . . . .      44    257    263    269    275    281
                      287    293    299    486    490
SQR . . . . . . . . . . . . .     310
SQRT  . . . . . . . . . . . .     263    275    287    299    310
STR . . . . . . . . . . . . .     353    354
STR1  . . . . . . . . . . . .     350    353    356
STR2  . . . . . . . . . . . .     350    354    355    356
STRINGTYPE  . . . . . . . . .      27     90
SV  . . . . . . . . . . . . .     305    307    308    309    310
SYSTEMCHECK . . . . . . . . .      96    463
TEXT  . . . . . . . . . . . .      52
THETA . . . . . . . . . . . .      40    230    234    237    238    239
                      243    244    249    255    261    267    273    279
                      285    291    348    353    424    425    431    492
TRUE  . . . . . . . . . . . .     129    154
TR_SLOT . . . . . . . . . . .      12    372    383    392
UNUSED  . . . . . . . . . . .     113    150    157
UPCASE  . . . . . . . . . . .     209    477    500
VAL . . . . . . . . . . . . .      61     67     69     74     76
VALID . . . . . . . . . . . .     111    116    126    129    146    147
                      155
VELOCITY  . . . . . . . . . .      43    213    433    474    486    490
WRITE . . . . . . . . . . . .      93    203    206    212    229    242
                      415    439    449    455    492    497
WRITELN . . . . . . . . . . .      86    135    165    166    168    169
```

```
                                180      181      182      183      184      185      186      187
                                188      189      190      191      192      193      194      195
                                196      197      198      199      200      201      202      223
                                225      226      321      322      323      430      431      432
                                433      434      442      467      470      502
```

APPENDIX E

A MODEL FOR THE PROPAGATION OF SHORT PULSES OF ULTRASOUND IN A SOLID

A model for the propagation of short pulses of ultrasound in a solid

John P. Weight
The City University, School of Electrical Engineering and Applied Physics, Northampton Square, London EC1V 0HB, England

(Received 9 July 1986; accepted for publication 29 November 1986)

A simple model has been introduced which allows calculations to be made of the particle displacement and velocity waveforms which arise when short pulses of ultrasound are radiated into solid media from transducers typical of those used in nondestructive evaluation. The model makes use of the plane- and edge-wave nature of the pulses radiated and, for simplicity, incorporates an empirical extension of the impulse response method for propagation in a fluid medium. Calculations made using the model are compared with other theoretical results obtained previously by harmonic synthesis, as well as with experimental measurements of pulse waveforms made using a miniature piezoelectric receiving probe. The results obtained are of particular relevance to the characterization of small targets using ultrasound.

PACS numbers: 43.20.Bi, 43.20.Fn, 43.35.Cg, 43.35.Zc

INTRODUCTION

The propagation of short pulses of ultrasound in a solid medium is both more complicated to formulate theoretically and more difficult to investigate experimentally than it is for the case of a fluid medium of propagation. Because of this, most studies of diffraction effects, including those that arise when short pulses are radiated from transducers used in nondestructive testing (NDT), have been confined to propagation in fluids. For instance, the variation of pressure waveform with field-point position in a fluid and the form of the corresponding pressure-amplitude beam profile are well understood theoretically using an impulse response approach and assuming idealized piston behavior of the source.[1–5] Recent experimental measurements in fluid media[4–6] have confirmed that similar diffraction effects occur in the field of practical NDT transducers. Where there are discrepancies, these are explained as a consequence of the nonideal behavior of real transducers.[7–10]

In solids, where both longitudinal (compression) and transverse (shear) body waves can propagate, a direct impulse response approach becomes far more difficult to apply. Using the Cagniard de Hoop[11] method, Aulenbacher and Langenberg[12] have recently given the impulse response, and hence the pulsed directivity pattern, of a line or point source radiating into a solid half-space. By suitable numerical integration techniques, they have extended the line source result to calculate the impulse response of an infinite ribbon source. Similarly, it is feasible that the impulse response of a circular source could be obtained numerically from the corresponding result for a point or annular source. However, as yet no such results seem to have been published.

An alternative method for calculating pulsed waveforms in solids has been adopted by Kawashima,[13] who has numerically evaluated the integral expressions[14] for the displacement amplitude for any point in the field of a circular source undergoing continuous sinusoidal motion. He then obtained pulsed displacement waveforms by harmonic synthesis.

A third approach is to make use of a finite-difference formulation to calculate the time development of displacements within a solid half-space. By inserting suitable surface boundary conditions into the elastic wave equation expressed in cylindrical coordinates, displacements due to a circular source undergoing arbitrary motion can be calculated.[15]

A disadvantage of all of these techniques is that they require considerable computation time on even a mainframe digital computer.

The philosophy of the present work is to introduce a simple model for the impulse response and to investigate its usefulness in making rapid, approximate calculations of pressure, particle velocity, and displacement waveforms. In common with existing methods to study the propagation of ultrasonic pulses in solids, the model assumes that the source behaves as a perfectly rigid indenter of constant diameter. The waveforms predicted by the model are compared with other calculations made using the finite-difference method[15] and with the recent results obtained by Kawashima[13] using numerical integration. The model's predictions are also compared with experimental observations of field-point waveforms made using a specially constructed miniature receiver. Further experimental results show stroboscopic visualizations of pulsed ultrasonic waves propagating in a transparent solid. These are particularly useful for illustrating the underlying principles of the model.

I. THEORY

A. Impulse responses in a fluid medium: Plane and edge waves

In order to develop the model, it is convenient to first review briefly the structure of the pulsed field in fluids. As described in greater detail in earlier work,[4,5] good physical understanding of the structure of the pulsed field of, for instance, a circular source radiating into a fluid is provided by the concept of a direct plane wave that travels in the geometrical beam region straight ahead of the transducer, together with a diffracted edge wave that propagates as a toroidal wave from the edge of the transducer. The edge wave does not have a uniform amplitude with direction of propagation, being stronger in the direction straight ahead of the rim than at angles on either side. Furthermore, the phase of the edge wave shifts by π as the direction of propagation goes through the normal from the edge. These characteristics serve to remind us that the edge wave is not just a simple wave from the rim; it arises, as does the plane wave, from the summation of Huygensian wavelets from the whole surface of the source. At points on the axis, the edge-wave components from all elements of the rim arrive simultaneously to give a pressure waveform that is an inverted replica of the earlier-arriving plane-wave pulse. Off axis, two major edge-wave contributions are observed: one from the nearer parts of the transducer rim, the other from the further points of the rim. Close to the transducer, the paraxial plane- and edge-wave pulses may be time separated. Further away, they increasingly overlap to give a time-differentiated pulse that decreases in amplitude with range from the transducer.

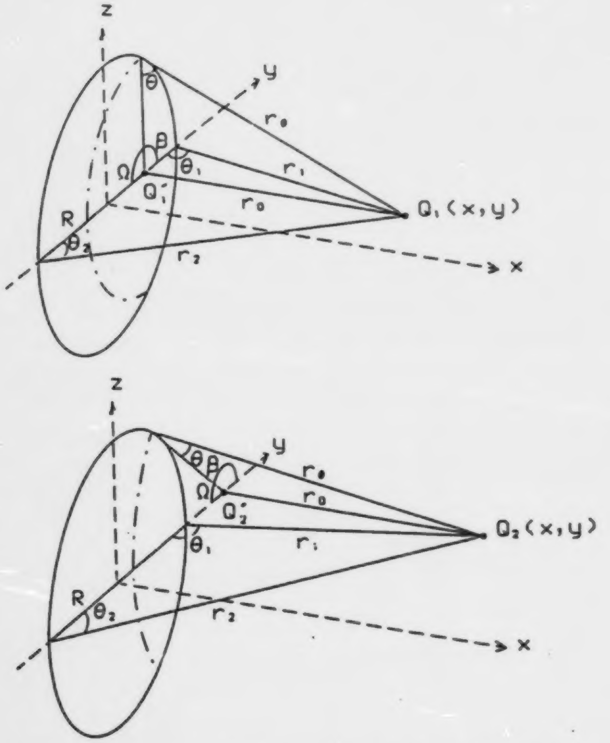


FIG. 1. Geometry for a circular source of radius R showing the angle θ from the circumference to field points which lie inside (Q_1) and outside (Q_2) the geometric beam. Q'_1 and Q'_2 are the perpendicular projections of Q_1 and Q_2 on to the plane of the source, and r_θ is the distance from the circumference to the field point. The limits of θ and r_θ are given by $\theta_1<\theta<\theta_2$, and $r_1<r_\theta<r_2$, respectively.

TABLE I. Expressions for the angle of equidistant arc (2Ω) on the surface of a circular source (after Robinson *et al.*[2]). The axes and other variables are defined in Fig. 1.

Region	Time limit	$\Omega(ct)$
Inside geometrical	$t_0<t<t_1$	π
beam, $y<R$	$t_1<t<t_2$	$\cos^{-1}\left(\dfrac{c^2t^2-x^2+y^2-R^2}{2y(c^2t^2-x^2)^{1/2}}\right)$
On edge, $y=R$	$t=t_0=t_1$	$\pi/2$
	$t_1<t<t_2$	$\cos^{-1}\left(\dfrac{(c^2t^2-x^2)^{1/2}}{2R}\right)$
Outside geometrical	$t_0<t<t_1$	0
beam, $y>R$	$t_1<t<t_2$	$\cos^{-1}\left(\dfrac{c^2t^2-x^2+y^2-R^2}{2y(c^2t^2-x^2)^{1/2}}\right)$

where

$t_0=x/c,$
$t_1=(1/c)[(R-y)^2+x^2]^{1/2},$
$t_2=(1/c)[(R+y)^2+x^2]^{1/2}.$

Quantitative calculations of field-point pressure waveforms that show this plane- and edge-wave structure may be made using the impulse response method.[1–3] The velocity potential impulse response ϕ_δ at a point (Q) (Fig. 1) in the field of an idealized piston source undergoing uniform motion and radiating into a lossless fluid medium is given by[1–3]

$$\phi_\delta(r,t)=c\Omega(ct)/\pi\,, \tag{1}$$

where $r=ct$ is the distance from the field point Q to equidistant points (an arc[4]) on the piston surface, $\Omega(ct)$ is the (half) angle of equidistant arc included on the surface, and c is the velocity of sound in the propagating medium. For the case of a circular source, analytic expressions for $\Omega(ct)$ have been given by a number of authors.[1,2,16] Those tabulated by Robinson *et al.*[2] are summarized in Table I for convenience.

The pressure impulse response p_δ at Q is given by

$$p_\delta(r,t)=\rho\left(\frac{\partial\phi_\delta}{\partial t}\right), \tag{2}$$

where ρ is the density of the fluid. The pressure waveform for arbitrary motion of the source $v(t)$ is then

$$p(r,t)=v(t)*p_\delta(r,t)\,, \tag{3}$$

where, as usual, $*$ denotes a convolution.

The plane- and edge-wave structure of the pulse radiated from a circular source can be clearly seen in the impulse responses given in Fig. 2. Figure 2(a) shows waveforms for an axial point. The impulse response then consists of two delta functions, the second (labeled e_l for compression edge) being an inverted replica of the first (labeled p_l for compression plane). [For convenience, throughout this article subscripts l and t denote compression (longitudinal) and shear (transverse), respectively.] After convolution with the source velocity function—here for clarity we choose a single-cycle sinusoid—these two delta functions give rise to the plane- and edge-wave pressure waveforms denoted P_l and E_l, respectively. The on-axis pressure impulse response may be written as

$$p_{\delta ax}(r,t)=\rho c_l[\delta(t-\tau_0)-\delta(t-\tau_l)]\,, \tag{4}$$

where c_l is the velocity of the compression wave in the propagating medium. The time $t=0$ is taken as the instant when the source first begins to move and τ_0 and τ_l are the arrival times at the field point for waves traveling from the center

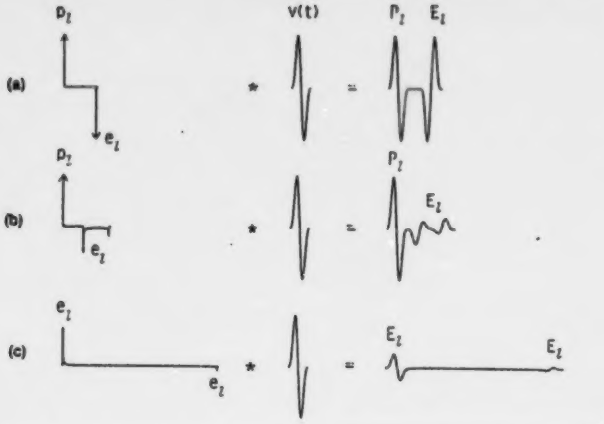


FIG. 2. Impulse responses and pressure waveforms at points at equal ranges from a circular source radiating into a fluid: (a) on axis; (b) off axis in the geometric region straight ahead of the source; and (c) outside the geometric region.

and the rim of the source, respectively.

Figure 2(b) shows the impulse response just off axis but within the geometric region straight ahead of the source. The pressure waveform now consists of a plane-wave component and two smaller, distorted edge-wave components. Finally, Fig. 2(c) shows waveforms for a point outside the geometrical region. Here, there is no plane-wave component, and the edge-wave component from the nearer edge is now inverted compared to the edge wave from the further edge.[4,5]

The waveforms shown in Fig. 2 are all at the same close range to the source. Further away, the impulse response waveforms are similar except that the time separation between each component becomes less. At a given increased range, the axial plane- and edge-wave components of the pressure waveform will overlap if $\tau_l - \tau_0 = \tau_d$ is less than the duration τ_v of the source velocity function $v(t)$. If $\tau_d \ll \tau_v$, the axial pressure waveform approximates to the time derivative of $v(t)$ since the impulse response then effectively approximates to the doublet function.

B. Model for the impulse response in a solid

Classical plane-wave theory predicts that a compression wave obliquely incident on a solid surface partially mode converts into a shear wave. Photoelastic visualization of bounded plane waves from real transducers confirms that this is so, but also shows the existence of a shear wave radiating from a normally coupled compression wave transducer. A number of other authors have also reported such behavior.[17–21]

Hayman[18] and Weight[17] have suggested that these shear waves are mode-converted edge waves and thus may be considered to be shear edge waves. This description is clearly illustrated in video recordings made by the present author. These were obtained for pulses propagating in fused quartz, using both directly and water-coupled transducers. By slowly increasing the delay before the stroboscopic light source is fired, the pulses could be followed out from the transducer, thus allowing the origin of the shear wave to be ascertained. Figure 3 shows a few still frames taken from such a recording, obtained using a directly coupled lead metaniobate (PMN) transducer. Since the clarity of the photographic reproduction is limited, the various wave fronts are identified schematically. The existence of the shear edge waves, as well as the expected compression plane and edge waves, is evident and must of course be taken account of in any model of pulse propagation in solids.

The basis of the present model is to assume that the compression edge wave that propagates from the rim of the transducer partially mode converts into a shear edge wave in proportion, depending on the angle from the field point, to each element of the source rim. This is achieved by "splitting" the compression edge-wave component from each element of the source rim into two components, one propagating at the compression-wave velocity, the other at the shear-wave velocity.

It is interesting to note here that close examination of photoelastic visualization results obtained using circular PMN transducers does not show the existence of any significant "head" waves similar to those reported in earlier work concerned with the propagation of pulses in fluid media.[7–10] However, a *shear* head wave can be seen in visualizations of the waves radiating from lead zirconate titanate (PZT) transducers, which are known to give rise to relatively strong

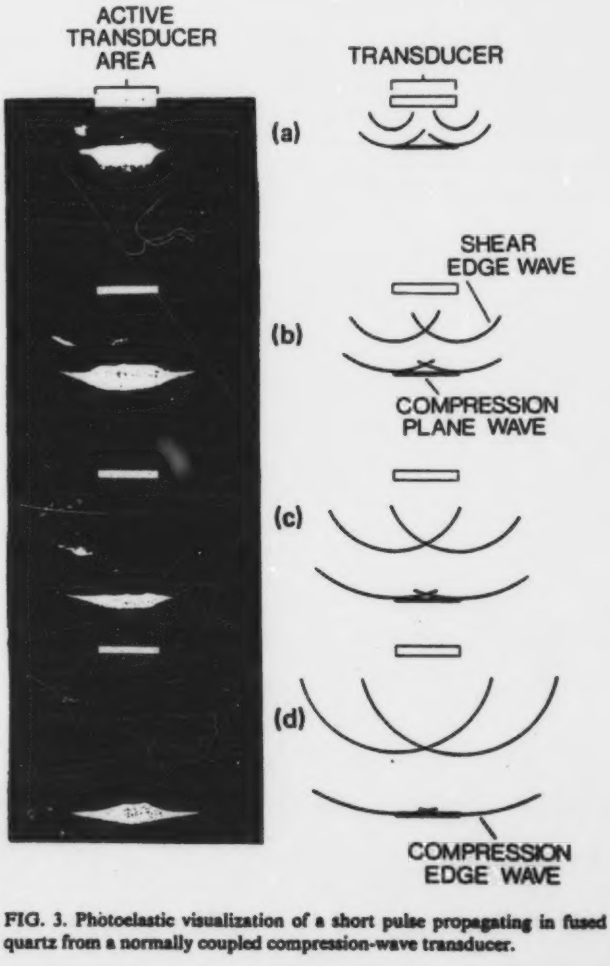


FIG. 3. Photoelastic visualization of a short pulse propagating in fused quartz from a normally coupled compression-wave transducer.

head waves,[4,7] and it is likely that weak head waves are in fact radiated from PMN transducers. A detailed discussion of the significance of head waves to the overall field structure of PMN and PZT transducers radiating into a solid is left for future work, but briefly, for most solid materials, their effect on field-point waveforms will only be of importance to the shear-wave component, and will only be of appreciable amplitude for points lying in a small paraxial region close to the transducer. Although the present model (in common with the other theoretical approaches outlined above) does not take account of the existence of head waves, it does nevertheless lead to reasonable agreement between calculated field-point waveforms and measured waveforms obtained using a PMN transducer.

C. On-axis field points

As an aid to introduce the model, it is helpful initially to restrict the discussion to points lying along the axis of the source. Then, the scalar pressure waveforms predicted by the model are as given in Fig. 4. Here, as is common, the "pressure" of a shear wave is defined as the shear force per unit area parallel to the direction of propagation. In Fig. 4, the (on-axis) impulse response now consists of three delta functions: p_l represents the plane-wave component and e_l and e_t the compression- and shear-wave components, respectively. Close to the source [Fig. 4(a)], the corresponding plane- and edge-wave pressure waveforms are time separated, but further away [Fig. 4(b) and (c)] the compression plane and edge waves overlap. However, the shear edge wave becomes further separated due to its lower velocity of propagation. The present model gives the axial pressure impulse response as

$$p_{\delta ax} = \rho c_l [\delta(t-\tau_0) - m_l(\theta)\delta(t-\tau_l) - c_t/c_l m_t(\theta)\delta(t-\tau_t)] , \tag{5}$$

where c_t is the shear-wave velocity, τ_t is the arrival time of the shear-wave component, and $m_l(\theta)$ and $m_t(\theta)$ are dimensionless mode-conversion factors that give the relative amplitudes of the particle velocities of the compression and shear edge waves, respectively. They are normalized as discussed below in order to relate the amplitudes of the edge-wave components to that of the compression plane wave. The angle θ is defined in Fig. 1, from which it can be seen that at a given range θ is a constant for points lying on axis. Then, of course, $m_l(\theta)$ and $m_t(\theta)$ are also constant.

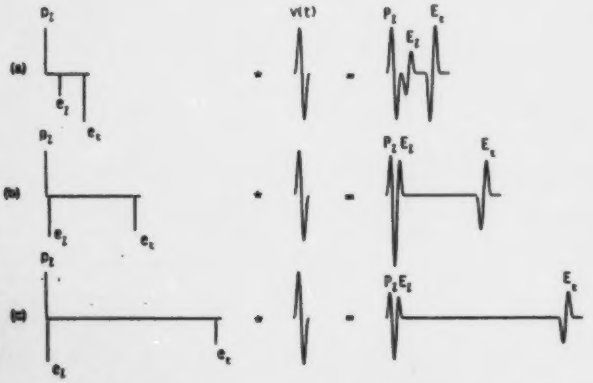


FIG. 4. Model predictions for pressure impulse responses (left) and waveforms (right) at various points lying on the axis of a circular source coupled to a solid propagating medium: (a) at short range, showing time-separated compression plane (P_l), compression edge (E_l), and shear edge (E_t) pulses; (b) and (c) at greater ranges, showing superimposed compression plane- and edge-wave pulses, but well separated shear edge-wave pulses.

D. Mode conversion

Before extending the discussion to deal with off-axis points, it is convenient to consider the angular variation of the mode-conversion factors $m_l(\theta)$ and $m_t(\theta)$. Although at first sight it might seem that a suitable assumption would be to consider that the incident compression edge wave is locally plane and to use classical plane-wave refraction expressions for $m_l(\theta)$ and $m_t(\theta)$, an examination of the pulse visualizations given in Fig. 3 exposes a weakness of such an approach. As can be seen in Fig. 3, there exists a component of the shear edge wave in the direction straight ahead of the edge of the source. A similar result is evident in finite-difference calculations made by Ilan and Weight[15] and reproduced here for convenience as Fig. 5. Plane-wave refraction

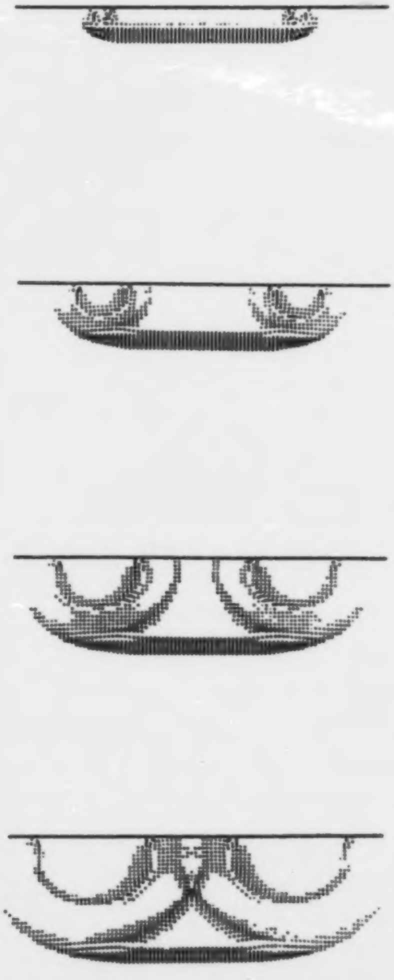


FIG. 5. Numerical visualization using the finite-difference method[15] to calculate particle displacements for a short pulse of ultrasound propagating from a circular source on the surface of a steel half-space.

theory predicts, of course, that no mode conversion occurs for a normally incident compression wave.

An alternative approach could be to consider that since the edge waves arrive from the periphery of the source, they could be modeled as a "folded" line source. This would permit the use of the above-mentioned directivity functions of Aulenbacher and Langenberg[12] to be used for the mode-conversion factors. However, Aulenbacher and Langenberg's directivity functions also show a zero amplitude for the shear wave straight ahead of the source, once again in conflict with experimental observations of shear *edge* waves.

The explanation for the existence of a shear edge wave straight ahead of the source rim lies in the complicated nature of the compression-wave displacements in that direction. As stated earlier, the polarity of the compression edge wave inverts as the angle from the rim passes through 90°. In addition, at field points just inside the geometric region, the compression plane- and edge-wave displacements arrive together and superimpose. As can be seen in finite-difference calculations, the resulting displacements do have a component in the direction perpendicular to the direction of propagation so that mode conversion can occur. Bearing in mind the philosophy of the present approach, it suffices to introduce simple empirical expressions for $m_l(\theta)$ and $m_t(\theta)$.

The functions used were chosen so that, when inserted into the impulse response model [Eq. (5)], the resulting edge-wave directivities were similar to those found using finite-difference calculations.[15] Figure 6 shows some polar plots of the mode-conversion functions used.

In Fig. 6, $m_l(\theta)$ is approximated by

$$m_l(\theta) = 1 - \exp(-a\theta) . \qquad (6)$$

The corresponding factor for the shear-wave component $m_t(\theta)$ is approximated by

$$m_t(\theta) = [b\theta \exp(1 - b\theta)]^2 , \qquad (7)$$

where θ, the angle from each element of the source rim to the field point, is given by

$$\theta = \sin^{-1}(r_0/r_\theta) ,$$

where r_0 and r_θ are the distances from the field point to the plane and each point of the rim of the source, respectively (Fig. 1), and a and b are normalizing factors to relate the amplitude of the edge-wave components to that of the compression plane wave. Values for a and b may be found by referring to finite-difference results[15] to obtain the ratio of the amplitudes of the two edge waves in the normal direction straight ahead of the source rim. For instance, for steel, the amplitude of the shear edge wave in the normal direction is 0.4 that of the compression edge wave. Assuming that in the normal direction, energy is conserved between the two edge waves,

$$m_l^2 + m_t^2 c_t/c_l = E_l^2 ,$$

where E_l is the amplitude of the "incident" compression edge wave. Thus, putting $E_l = 1$ and $m_t = 0.4 m_l$, then a in Eq. (6) becomes 2. The constant b may be found in a similar fashion, except that as a result of the form of Eq. (7), iteration is required. For steel, b comes out as 1.5. Finite-difference results have also been obtained for the ratio of the edge-wave amplitudes for pulses propagating in aluminum. These show that for the purposes of the present model similar values for a and b can be used for aluminum as those given above for steel.

Returning to Fig. 4, it is now possible to discuss how the form of the mode-conversion factors affects the amplitude of the edge-wave components at various ranges along the axis of the source. At short range [Fig. 4(a)], the shear edge-wave pulse is larger than the compression edge wave since the angle θ subtended back at the rim of the source is such that strong mode conversion occurs. At greater ranges (*b*,*c*), the mode conversion is less pronounced and the compression edge-wave pulse (which increasingly overlaps the plane-wave pulse) retains much of its energy.

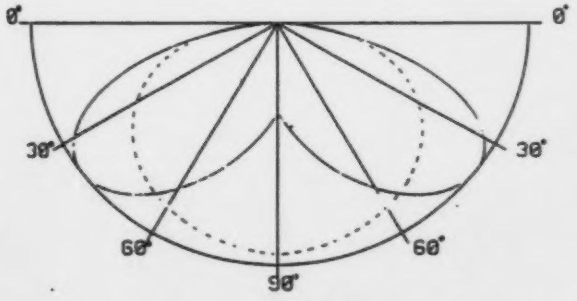


FIG. 6. Polar plots of the empirical "mode-conversion" factors $m_l(\theta)$ (dashed) and $m_t(\theta)$ (continuous).

E. Off-axis field points

For off-axis points, the distance between each element of the source (semi) circumference and the field point is different. As seen in Fig. 1, the angle θ subtended at each element also varies. The model assumes that the contribution from each element of the rim as predicted for a fluid propagating medium can be weighted according to $m_l(\theta)$ and $m_t(\theta)$ to give the off-axis edge-wave contributions for the solid case. The pressure impulse response may now be written as

$$p_8 = \rho c_l [\delta(t - t_0) - m_l(\theta) f_l(t - t_l) - c_t/c_l m_t(\theta) f(t - t_t)] , \qquad (8)$$

where $f_l(t - t_l)$ is the edge-wave contribution predicted for a fluid propagating medium having sound velocity c_l and $f_t(t - t_t)$ is that for a medium having sound velocity c_t. As described later, $f_l(t - t_l)$ and $f_t(t - t_t)$ are extracted from the impulse responses given by Eqs. (1) and (2) and the expressions listed in Table I. Thus $m_l(\theta) f_l(t - t_l)$ and $m_t(\theta) f_t(t - t_t)$ are the compression and shear edge-wave contributions, respectively, t_l and t_t being the arrival times at the field point of the compression and shear-wave contributions from each element of the source circumference. The limits of t_l and t_t are given by

$$r_1/c_l < t_l < r_2/c_l$$

and

$$r_1/c_t < t_t < r_2/c_t ,$$

where (Fig. 1) r_1 and r_2 are the distances from the field point to the near point and the far point of the source circumference, respectively. The angle θ is now given by

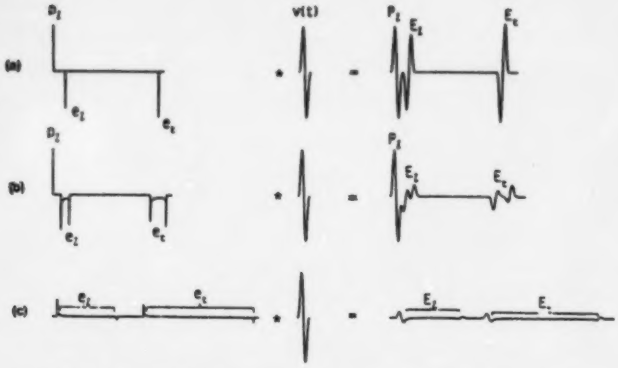


FIG. 7. Model predictions for pressure impulse responses (left) and waveforms (right) at various points at the same range close to the source: (a) on axis; (b) off axis within the geometric region; (c) outside the geometric region.

$$\theta = \sin^{-1}(r_0/c_l t_l) = \sin^{-1}(r_0/c_t t_t)\ . \tag{9}$$

Figure 7 shows pressure waveforms predicted by the model for points on and off axis at the same range. In the off-axis result shown in Fig. 7(b), the plane-wave pulse P_l is the same as that of the on-axis result [Fig. 7(a)], but each of the edge waves now consists of two time-separated pulses (E_l and E_t). Outside the geometric region [Fig. 7(c)], there is no plane wave: As in Fig. 7(b), each edge-wave pulse consists of two main components, one from the nearer edge of the source, the other from the further edge. From Eq. (8), the two edge-wave components of the impulse response are

$$e_l = \rho c_l m_l(\theta) f_l(t - t_l) \tag{10}$$

and

$$e_t = \rho c_t m_t(\theta) f_t(t - t_t)\ . \tag{11}$$

F. Vector particle velocities and displacements

So far, an extension of scalar propagation theory has been considered which gives the pressure at points in the field of a circular source coupled to a solid. However, in order to give a complete treatment for propagation in solids, vector particle velocities and displacements are required. This will allow a comparison of the model's predictions with experimental measurements made using miniature receiving probes which, as discussed below, respond to surface particle velocities. In addition, calculations of particle velocity and displacement waveforms may be compared with other results made using established cw theory and harmonic synthesis.[13]

The approach adopted is to make use of a knowledge of the plane- and edge-wave structure of the pulses radiated. This, together with the assumption that the wave fronts can be considered locally plane, will then allow vector particle velocities to be obtained as described below. Particle displacements are obtained by integration of the corresponding velocity waveforms.

Figure 8 shows a two-dimensional schematic representation of the plane and edge waves radiated, including their relative polarities. The convention adopted for the sign of compression- and shear-wave particle velocities is shown in the figure. The radial and tangential components of the edge-wave portions of the particle-velocity impulse response u_δ are denoted u_r and u_θ, respectively. The corresponding vertical and horizontal components of the compression-wave particle velocities in the plane $z = 0$ are denoted by u_{xl} and u_{yl}, respectively. Similarly, the vertical and horizontal components of the shear wave are u_{xt} and u_{yt}. The polarities indicated by the arrows representing the vector particle velocities are those predicted by scalar impulse response theory and their direction is as given by the plane- and edge-wave structure of the wave fronts radiated. Then, with θ and β as defined in Figs. 1 and 8,

$$u_{xl} = u_l \sin\theta\ , \tag{12}$$

$$u_{yl} = -u_l \cos\theta \cos\beta\ , \tag{13}$$

$$u_{xt} = u_t \cos\theta\ , \tag{14}$$

$$u_{yt} = u_t \sin\theta \cos\beta\ , \tag{15}$$

where, by assuming that the edge waves may be considered locally plane, the amplitudes of the compression and shear particle velocities are given by, respectively,

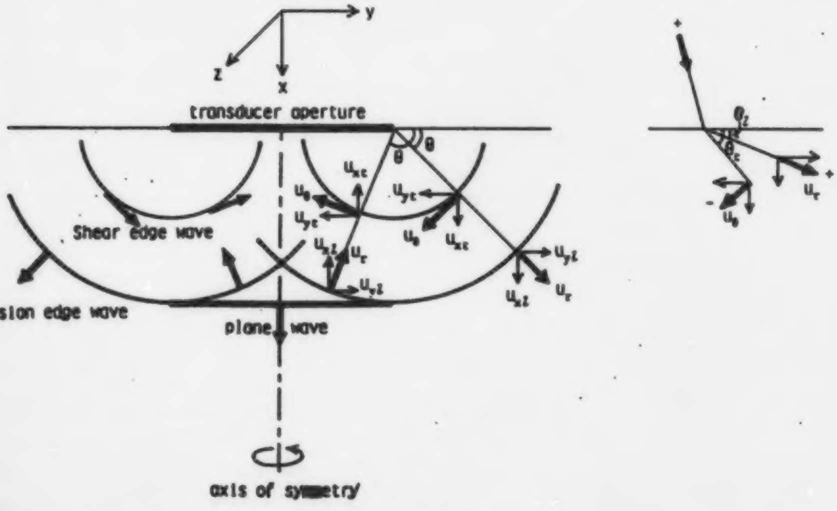


FIG. 8. Relative polarities of the plane and edge waves radiated from a circular source coupled to a solid. The inset shows the sign convention adopted for the mode conversion of an incident compression wave.

$$u_l = e_l/\rho c_l \tag{16}$$

and

$$u_t = e_t/\rho c_t\ , \tag{17}$$

where e_l and e_t are the compression and shear edge-wave impulse responses given in Eqs. (1)–(11) and Table I.

The particle-velocity impulse responses ($u_{x\delta}$ and $u_{y\delta}$) predicted for various points close to a circular source are shown in Fig. 9. The various pulses which comprise the vertical component $u_{x\delta}$ of the particle-velocity impulse response have the same polarity as those of the pressure impulse response. However, the phase relationship of the horizontal component $u_{y\delta}$ is more complicated. On axis, there is no component in the horizontal direction for either the plane or the two edge waves. For off-axis points within the geometric beam to the right of the axis of symmetry (i.e., for $0 < y < R$), the horizontal component of the compression edge wave from the *nearer* edge is inverted compared to the corresponding vertical component. Conversely, it is the horizontal component of the shear edge wave from the *further* edge that is now inverted. For off-axis points to the right of the geometric region (i.e., $y > R$), different polarity relationships are evident. The compression-wave horizontal and vertical components have the same polarity, but the corresponding shear-wave components are inverted. As indicated by the axis of symmetry, all *horizontal* components for points to the left of the beam axis (i.e., $y < 0$) are inverted compared to those of corresponding points to the right of the axis. However, the vertical components have the same polarity on either side of the beam axis.

Often, radial and tangential components are referred to the angle θ' from the center of the source. Here, these are denoted u'_r and u'_θ, respectively, and are given by

$$u'_r = (u_{xl} + u_{xt}) \cos\theta' + (u_{yl} + u_{yt}) \sin\theta'\ , \tag{18}$$

$$u'_\theta = -(u_{xt} + u_{xl}) \sin\theta' + (u_{yt} + u_{yl}) \cos\theta'\ . \tag{19}$$

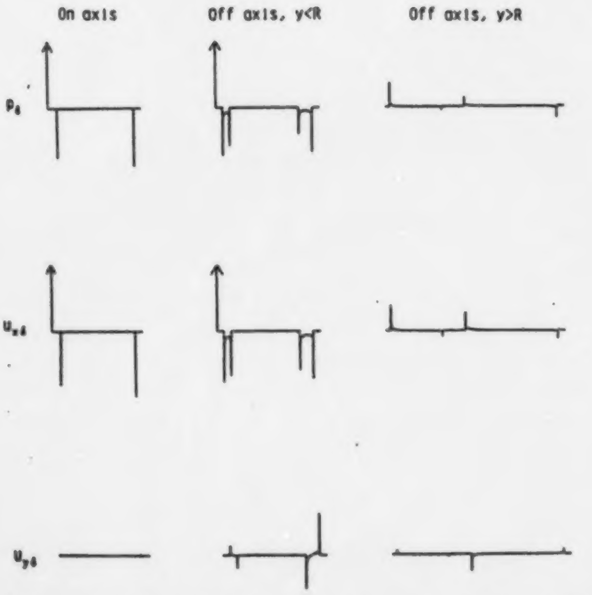


FIG. 9. Model predictions for pressure p_δ and particle velocity impulse responses (vertical component $u_{x\delta}$, horizontal component $u_{y\delta}$) at various points at the same close range to a circular source of radius R.

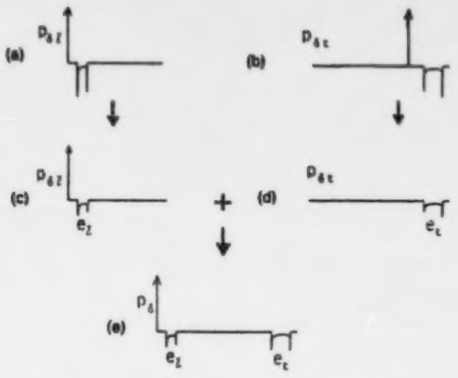


FIG. 10. Graphic representation of the computation steps used in calculating the approximate pressure impulse response for a circular source. The responses shown in (a) and (b) are concepts used in the model and have only limited physical reality.

II. CALCULATED AND MEASURED WAVEFORMS

A. Calculations using the model

The computational steps used in making waveform calculations are illustrated graphically in Fig. 10. For the point within the geometric region considered, the compression and shear "impulse responses" $p_{\delta l}$ and $p_{\delta t}$ will consist of a delta function representing a plane wave, followed by the later-arriving edge-wave contributions. Provided that the time increment used in the calculations is less than the time difference between the (compression) plane-wave contribution arriving at a field point and the (compression) edge-wave contribution, the plane and edge waves can easily be separated to be weighted as required by the model. In Fig. 10(c) the compression edge wave e_l is weighted according to Eqs. (6), (9), and (10). In (d) the shear "plane" wave has been discarded and the shear edge wave e_t weighted according to Eqs. (7), (9), and (11). The total pressure impulse response $p_\delta(r,t)$ is then simply the sum of (c) plus (d). For points lying outside the geometric region, the procedure is similar except that it is now no longer necessary to deal with plane-wave components.

Particle-velocity impulse responses can be obtained from the pressure impulse response using Eqs. (12)–(17). The pressure or particle-velocity waveforms for arbitrary motion of the source may then be found by numerical convolution of the relevant impulse response with the given source velocity function. Both numerical integration in the time domain and fast Fourier transform techniques have been used to perform the convolutions. For the source apertures, source velocity functions, and field-point positions investigated here, both methods give similar efficiency. Displacement waveforms may be obtained by numerical integration of the corresponding particle-velocity waveforms.

Figure 11 shows computed pressure and particle-velocity (vertical component) waveforms at various points in the field of a circular source of radius 8 mm radiating a single-

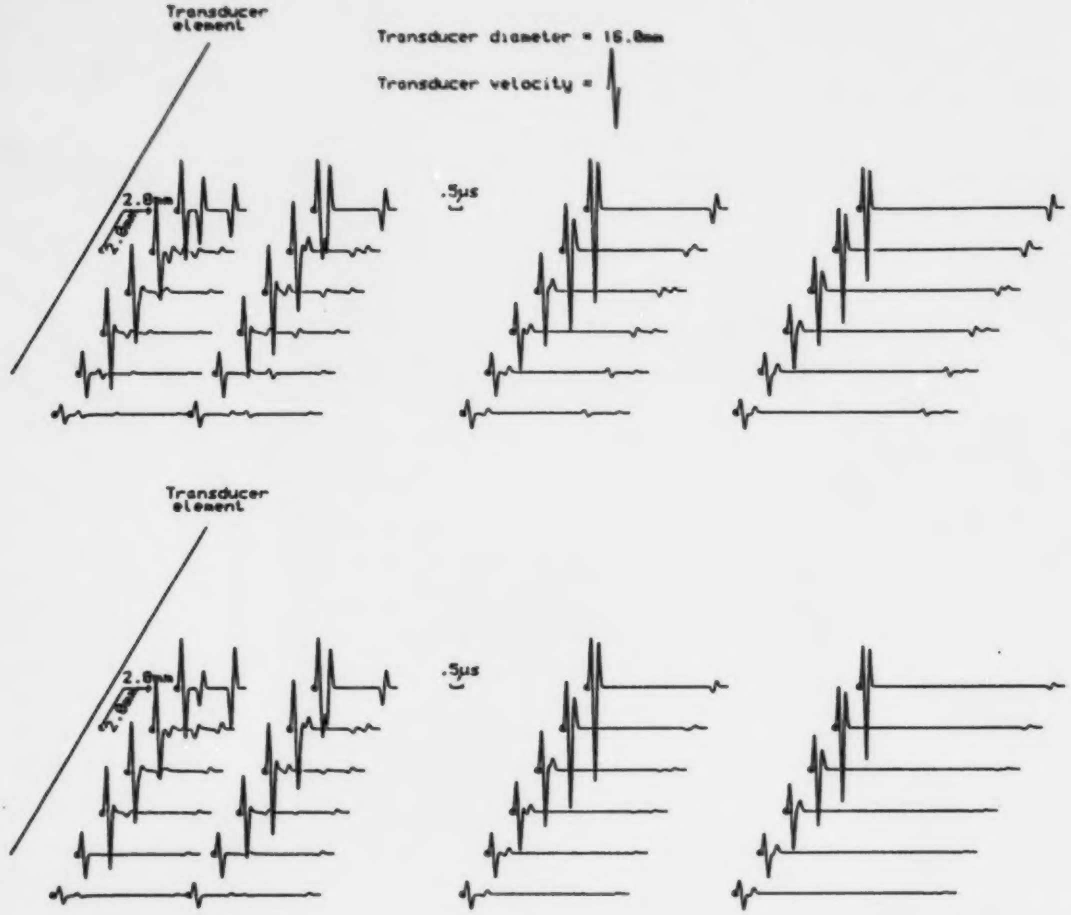


FIG. 11. Computed waveforms at points (●) in the field of a circular source of radius 10 mm radiating a short pulse into steel ($\rho = 7.8 \times 10^3$ kg m^{-3}, $c_l = 5.85 \times 10^3$ m s^{-1}, $c_s = 3.23 \times 10^3$ m s^{-1}). Top, pressure waveforms; bottom, vertical component of the particle-velocity waveforms.

cycle of a 2-MHz sine wave into steel. Close to the source, all of the plane- and edge-wave components can be clearly seen as time-separated pulses. Further away, the compression plane- and edge-wave components overlap—especially near the axis of the source—but the shear-wave component becomes further separated from the compression waves.

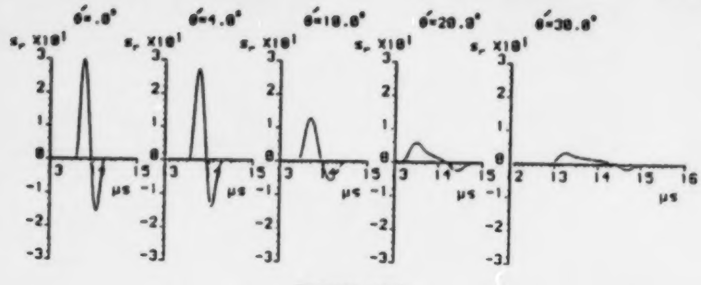
B. Comparison with harmonically synthesized results

Figure 12 shows a comparison of results calculated using the model with those calculated by Kawashima[13] for a 10-mm radius source radiating a short pulse into aluminum. Because these are at a range of 85 mm, there is a relatively large time separation between the compression- and shear-wave components. In order to use a time scale which allows the form of each component to be clearly seen, they are displayed separately in the same fashion adopted by Kawashima. Since the plane-wave pressure waveform generated by an ultrasonic source has the same form as the velocity of the source, the source velocity function used for the model calculations was a scaled version of Kawashima's function for the force generated by his (ceramic) transducer. The scaling coefficient was chosen so that the on-axis displacement waveforms (radial component) were of similar amplitude in both sets of results.

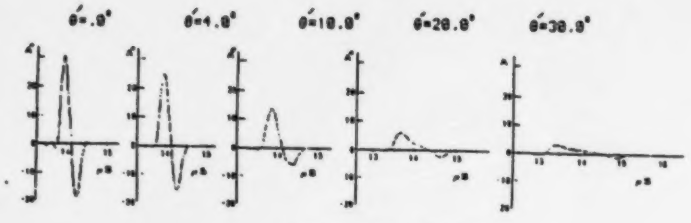
Kawashima also gives measured and calculated results for the output voltage waveforms generated by small (2-mm-sq) receiving probes. His calculated voltage waveforms are obtained by differentiating the displacement waveforms and, therefore, have the same form as the particle velocity. Figure 13 shows a further comparison of the model's predictions and Kawashima's results, this time for particle velocity and receiving transducer voltage waveforms. Again, the present calculations have been scaled so that the amplitudes of the on-axis particle velocity waveforms (radial component) are similar. In all cases there is good agreement between the two sets of results.

C. Field-point waveform measurements

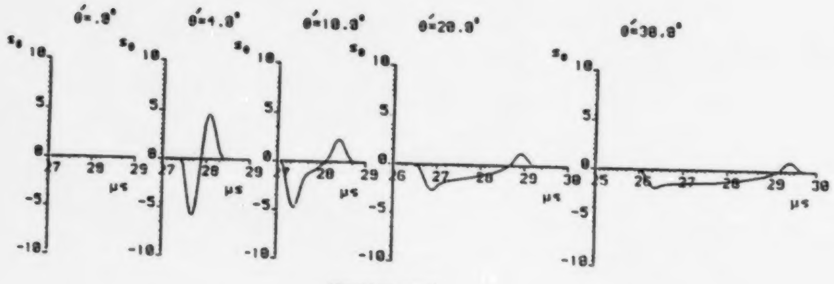
Previous experimental observations[4,5,7] made in fluids have shown that miniature (0.2-mm-diam) probes constructed using an active element of lead metaniobate may be used to measure pulsed ultrasonic waveforms having spectra extending from 0.5–8 MHz. Recently, it has been shown that the probes may be used to detect both compression and shear



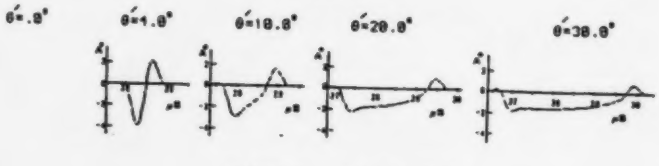
Present work



After Kawashima (1984)



Present work



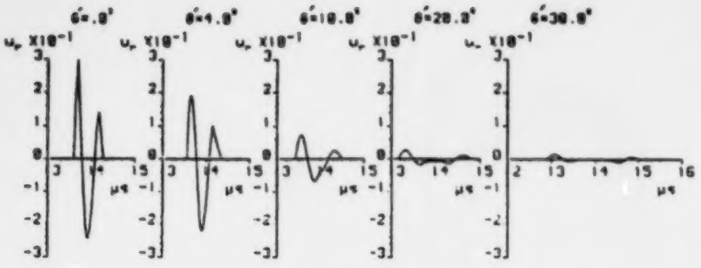
After Kawashima (1984)

FIG. 12. Comparison of displacement waveforms calculated using the present model with those obtained by Kawashima.[13] Top, compression-wave components; bottom, shear-wave components. The compression and shear waveforms are for the radial (s_r) and tangential (s_θ) displacements, respectively, at points distant 85 mm from a circular source of radius 10 mm coupled to aluminum ($\rho = 2.7 \times 10^3$ kg m^{-3}, $c_l = 6.2 \times 10^3$ m s^{-1}, $c_t = 3.1 \times 10^3$ m s^{-1}). Here θ' is the angle from the field point to the center of the source.

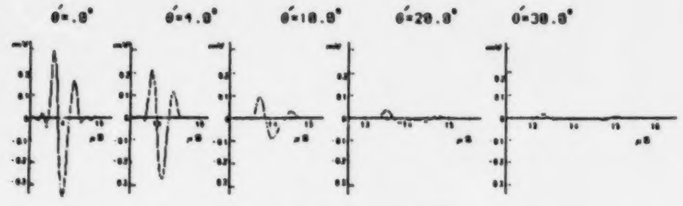
waves propagating from within a solid to impinge on a surface. A detailed account of the use of the miniature probes to measure short pulses propagating in solids will be given elsewhere,[22] but briefly, the action of the probes is such that they respond to the normal component of the surface velocity motion which is assumed to be locally plane. For simplicity, it is assumed that the surface acts as a free boundary which is not perturbed by coupling a miniature probe. Furthermore, as has been assumed by Kawashima,[13] the normal component of the surface particle velocity is taken to be twice the vertical component of the body wave incident on the surface. Although this is only approximately true (see, for instance, Gutenberg[23]), experimental measurements made to quantify the action of the miniature probes[22] show that such an assumption is reasonable. Thus the probes may be calibrated to measure both compression and shear waves provided that the incident angle to the surface is known and is such that a component in the direction normal to the surface exists. At frequencies up to about 8 MHz, the diameter of the probe is smaller than the incident wavelengths for most solids and may be considered to act as a "point" receiver.

D. Measured waveforms of pulses propagating in steel

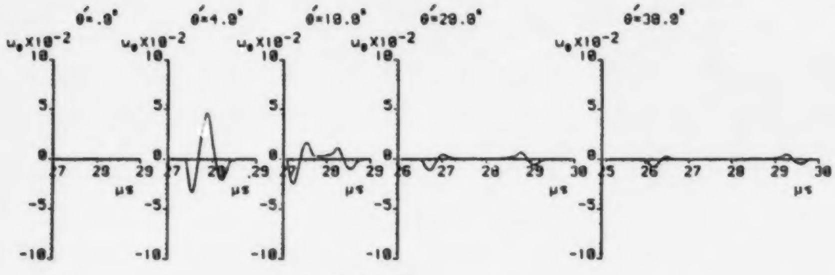
The waveforms appearing at the surface of steel test pieces of various thickness were measured using the miniature probe supported in a specially constructed jig which was used to avoid problems due to variations in coupling. The jig was clamped to the test piece and was not disturbed throughout the course of each set of measurements, all alignments being carried out by moving the transmitting transducer. Each block had a round-bottomed hole of 10-mm radius



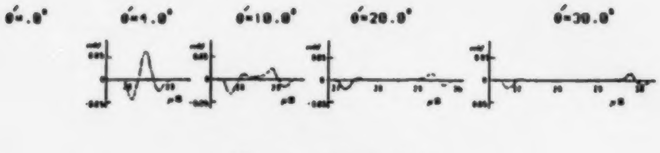
Present work



After Kawashima (1904)



Present work



After Kawashima (1984)

FIG. 13. Comparison of particle velocity waveforms (u_r and u_θ) calculated by the present model with those obtained by Kawashima[13] for the voltage output of an idealized particle-velocity sensitive receiver. Top, compression-wave components. Bottom, shear-wave components. The source configuration is the same as in Fig. 12.

bored into one surface and the tip of the miniature probe was coupled to the bottom of each hole (Fig. 14). The function of the hole was to avoid a problem which occurred with some block thicknesses due to overlapping of the shear edge wave with a multiple reflection (between the two surfaces of the block) of the faster-propagating compression wave. The transmitter—a 19-mm-diam lead metaniobate transducer (V3825, supplied by Panametrics)—was directly coupled to the opposite side of the test piece.

Figure 15 shows a typical set of results for mild steel. For comparison, a set of calculated results using the simple model is also given. Since, as mentioned in Sec. II C, the probe responds to the normal component of the surface velocity, the calculated results are also for the normal component of surface particle velocity. On axis, the pulse (E_t) arising from the shear edge wave predicted by the model is clearly resolved, as, at the shorter ranges, is the compression edge-wave pulse (E_l). The arrival time of each pulse in the measured results is as predicted and there is reasonable agreement of the relative amplitudes of the various pulses at each range. Off axis, the shear edge-wave pulse can be seen to split into two main components, one from the nearer portion of the source rim, the other from the further portion of the rim. The compression edge-wave pulse also splits into components, but for most of the field-point positions at which these results were taken, the compression plane and edge waves overlap, so that the two edge-wave components are less obvious. As anticipated, the compression plane-wave pulse only changes with position off axis where it is superimposed by a compression edge-wave pulse.

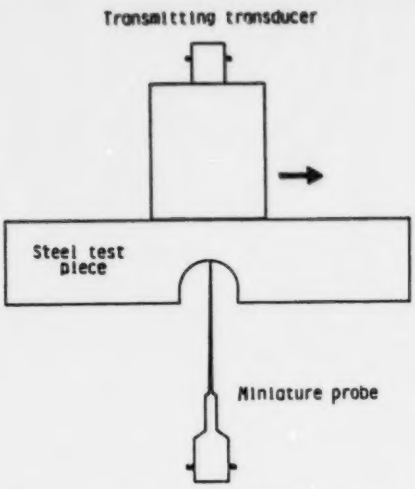


FIG. 14. Measurement of particle-velocity waveforms using a miniature piezoelectric receiver.

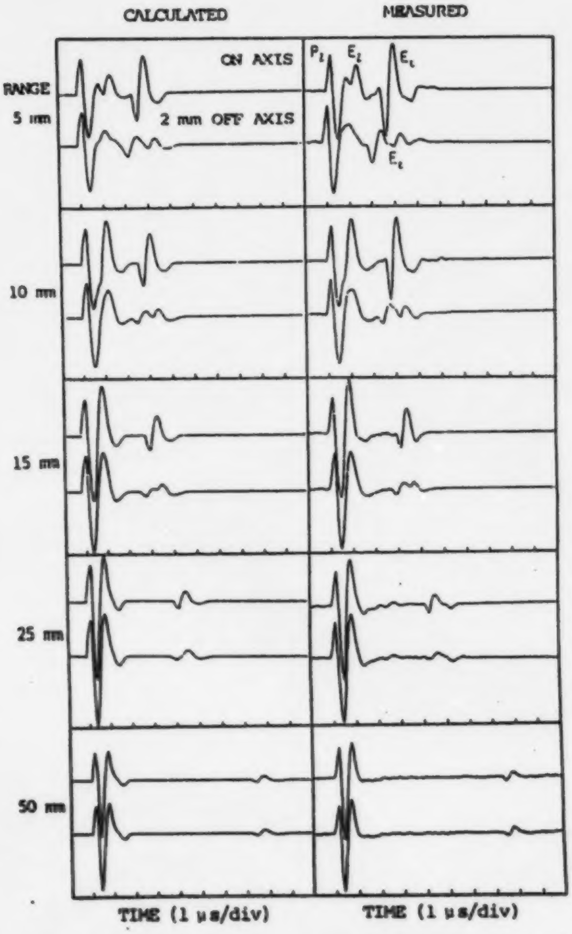


FIG. 15. Calculated and measured waveforms at points on the surface of steel test pieces of various thickness. The transmitted pulse was launched from a 19-mm-diam transducer (Panametrics V3289) directly coupled to the opposite surface of each test piece (see Fig. 14). Here P_l and E_l denote a compression plane- and edge-wave pulse, respectively, and E_t denotes a shear edge-wave pulse.

III. DISCUSSION

The results given in Figs. 12 and 13 show that calculations using the present model compare well with those calculated by Kawashima[13] using the harmonic synthesis approach. The relative amplitudes of each pulse agree quite well and the time position of each pulse is in good agreement to that of Kawashima, except that the arrival time of the compression plane wave in his on-axis result indicates a path of 86 mm instead of the 85 mm quoted. The arrival times of the pulses at points lying outside the geometrical beam region agree well and are consistent with the argument that the time origin of such pulses is at the rim of the source. In addition, the phase relationships between the radial and tangential components and their variation with field-point position are the same in both sets of results.

The form of the displacement waveforms for field points lying outside the geometric region may be explained thus: In Fig. 5, the off-axis edge-wave particle-velocity waveforms are not symmetric about the time axis, even for the symmetric source velocity function used. This results from the form of the off-axis edge-wave component of the impulse response, which is not a simple delta function but is spread out in time as shown in Fig. 7. Thus the displacement waveform (given by the integral of the velocity waveform) has a relatively large amplitude between the arrival time of a component from the nearest element of the source rim and the arrival time of the last component from the furthest point of the rim. The corresponding particle-velocity waveform has a "split pulse" form similar to that observed previously[1–10] in work concerned with propagation in fluids. In earlier work,[4,5,8–10] this pulse splitting was explained as being due to the plane- and edge-wave structure of the pulses radiated. Kawashima[13] doubts this, but it would seem that the present results lend further support for such an explanation. Not only can the plane and edge waves be clearly seen in the photoelastic visualizations given in Fig. 3, but also they are evident in the finite-difference numerical visualization shown in Fig. 5. In addition, as pointed out above, there is good agreement between calculated results made using a model based on the concept of plane and edge waves and those obtained independently by Kawashima.

The plane- and edge-wave structure of the pulses also explains the phase relationships between the vertical and horizontal components at various positions in the field. For instance, as seen in Fig. 8, the on-axis horizontal components of both the compression and the shear edge waves arriving from diametrically opposed elements of the rim are equal and opposite. Thus, on axis, the horizontal components of both edge waves are zero. However, as also seen in Fig. 8, the on-axis vertical components all add to give a relatively large contribution. Similarly, Kawashima's voltage waveforms (i.e., particle-velocity waveforms) for off-axis points clearly show the same phase relationships as predicted by the present model and illustrated in Figs. 8 and 9.

The variation of the amplitude of the horizontal and vertical components with field position is obviously a major factor in deciding which kind of detector to use when experimental waveform measurements are to be made at various field points. Using electromagnetic acoustic transducers, it is possible to detect both components,[13] but most detectors respond to the vertical component. Fortunately, for the paraxial region of major interest, it is best to use a detector which responds to the vertical component of the waveform, assuming of course that the surface at which detection takes place is parallel to that from which the pulses are propagated.

IV. CONCLUSION

Stroboscopic photoelastic visualization of short ultrasonic pulses propagating in an isotropic solid show that both compression and shear waves are propagated from a normally coupled compression-wave transducer. The compression waves radiated by a circular lead metaniobate transducer coupled to a solid are similar to those for the case of a fluid propagating medium and consist of a plane wave propagating in the geometric region straight ahead of the source, together with a toroidal edge wave which propagates from the rim of the source. In contrast, the shear wave consists primarily of an edge wave alone—there being no shear plane wave, but some evidence of a weak shear head wave. A similar plane- and edge-wave structure is seen in finite-difference calculations of the displacements radiating from a circular source coupled to a solid half-space.

A simple model which allows rapid approximate calculations of particle-velocity and displacement waveforms to be made explains the origin of the shear edge wave as being due to the partial mode conversion of the compression edge wave. Calculations made using the model agree with other calculations using a harmonic synthesis approach[13] and with measured field-point waveforms obtained using specially constructed miniature receiving probes.

As pointed out in earlier work concerned with the propagation of short ultrasonic pulses in fluids,[4,5,7,17] a multipulse structure arises due to the existence of plane and edge waves. For pulse-echo testing this structure complicates the interpretation of the echo responses from small targets. In solids, the existence of the shear edge wave further complicates the interpretation of such results. Fortunately, at target ranges greater than a few transducer radii, the shear-wave component becomes well separated in time from the compression wave and can often be time gated.

[1] P. R. Stepanishen, "Transient radiation from pistons in an infinite planar baffle," J. Acoust. Soc. Am. **49**, 1629–1638 (1971).

[2] D. E. Robinson, S. Lees, and L. Bess, "Near field transient radiation patterns for circular pistons," IEEE Trans. Acoust. Speech Signal Process. **ASSP-22**, 395–403 (1974).

[3] G. R. Harris, "Review of transient field theory for a baffled planar piston," J. Acoust. Soc. Am. **70**, 10–20 (1981).

[4] J. P. Weight and A. J. Hayman, "Observations of the propagation of very short ultrasonic pulses and their reflection by small targets," J. Acoust. Soc. Am. **63**, 396–404 (1978).

[5] J. P. Weight, "Ultrasonic beam structures in fluid media," J. Acoust. Soc. Am. **76**, 1184–1193 (1984).

[6] P. R. Stepanishen and G. Fisher, "Experimental verification of the impulse response method to evaluate transient acoustic fields," J. Acoust. Soc. Am. **69**, 1610–1617 (1981).

[7] A. J. Hayman and J. P. Weight, "Transmission and reception of short ultrasonic pulses by circular and square transducers," J. Acoust. Soc. Am. **66**, 945–951 (1979).

[8] M. R. Layton, E. F. Carome, H. D. Hardy, and J. A. Bucaro, "Effects of diffraction on stress pulse propagation," J. Acoust. Soc. Am. **64**, 250–256 (1978).

[9] G. R. Harris, E. F. Carome, and H. D. Hardy, "An analysis of pulsed ultrasonic fields as measured by PVDF spot-poled membrane hydrophones," IEEE Trans. Sonics Ultrason. **SU-30**, 295–303 (1983).

[10] J. C. Baboux, F. Lakestani, and M. Perdrix, "Theoretical and experimental study of the contribution of radial modes to the pulsed ultrasonic field radiated by a thick piezoelectric disk," J. Acoust. Soc. Am. **75**, 1722–1731 (1984).

[11] A. T. de Hoop, "A modification of Cagniard's method for solving seismic wave problems," Appl. Sci. Res. **B8**, 349–365 (1959).

[12] U. Aulenbacher and K. J. Langenberg, "Transient radiation pattern of ultrasonic transducers on elastic half-spaces," in *New Procedures in Nondestructive Testing*, edited by P. Holler (Springer, Berlin, Heidelberg, 1983), pp. 81–92.

[13] K. Kawashima, "Quantitative calculation and measurement of longitudinal and transverse ultrasonic wave pulses in solid," IEEE Trans. Sonics Ultrason. **SU-31**, 83–94 (1984).

[14] G. F. Miller and H. Pursey, "The field and radiation impedance of mechanical radiators on the free surface of a semi-infinite isotropic solid," Proc. R. Soc. London Ser. A **223**, 521–541 (1954).

[15] A. Ilan and J. P. Weight, "The propagation of short ultrasonic pulses from a circular source coupled to a solid half-space" (to be published).

[16] F. Oberhettinger, "On transient solutions of the baffled piston problem," J. Res. Nat. Bur. Stand. **65B**, 1–6 (1961).

[17] J. P. Weight, "The propagation and reception of wide-band ultrasonic pulses," Ph.D. thesis, The City University, London (1982).

[18] A. J. Hayman, "Schlieren visualization of focused ultrasonic images," Ph.D. thesis, The City University, London (1977).

[19] K. G. Hall, "A qualitative evaluation of variable-angle ultrasonic transducers by the photoelastic visualization method," Ultrasonics **15**, 245–252 (1977).

[20] W. Sachse, N. N. Hsu, and D. G. Eitzen, "Visualization of transducer-produced sound fields in solids," IEE Symposium Proc. 139–143 (1978).

[21] C. F. Ying, M. X. Li, and H. L. Zhang, "Computations and measurements of transient stress waves and electrical voltages generated by transmitting ultrasonic piezoelectric transducers," Ultrasonics **19**, 155–158 (1981).

[22] J. P. Weight, "Measurement of longitudinal and transverse waves of pulsed ultrasound in solid media," to be published in Ultrasonics.

[23] B. Gutenberg, "Energy ratio of reflected and refracted seismic waves," Bull. Seismol. Soc. Am. **34**, 85–102 (1944).

DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLLED TWO-DIMENSIONAL ACOUSTIC ARRAY FOR IMAGE DEFECTS IN OFFSHORE STRUCTURE WELDMENTS

PHASE 2

WORK ON THIS PROJECT WAS FUNDED BY THE METALS TECHNOLOGY LABORATORIES, CANADA CENTRE FOR MINERAL AND ENERGY TECHNOLOGY, ENERGY MINES AND RESOURCES CANADA.

DSS FILE NUMBER: 03SQ.23440-5-9165

DSS CONTRACT SERIAL NUMBER: OSQ85-00304

CANMET SCIENTIFIC AUTHORITY: DR. D.K. MAK

SUBMITTED TO:

ENERGY, MINES AND RESOURCES
CANMET, METALS TECHNOLOGY LABORATORIES
568 BOOTH STREET
OTTAWA, ONTARIO
KIA OG1

SUBMITTED BY:

FLEET TECHNOLOGY LIMITED
311 LEGGET DRIVE
KANATA, ONTARIO
K2K 1Z8

JANUARY, 1990

Abstract

This report includes both the first and second phases of the design and construction of a 2-D acoustic array for imaging defects in steel weldments. Phase I covers the design and construction of the array and associated electronics. Phase II covers an improved array manufacturing process, the image reconstruction procedure, and the software development. Hardware problems prevented acquisition of useful test data, therefore, no results are presented. Loss of key personnel contributed to the discontinuation of the project.

Résumé

Ce rapport décrit les deux phases effectuées pour la conception et la fabrication d'un étalage acoustiques en deux dimensions pour reproduire l'image des defectuosités dans les soudures de métal.

La première phase couvre la conception, et la construction d'un étalage acoustique et d'un système éléctronique associé. La seconde phase couvre le development d'un procédé de fabrication amélioré, les procédures pour obtenir la reconstruction de l'image et le development du logiciel associé.

Des problèmes recontrés avec le système n'ont pas permis la collection de données des éssais utiles, pour cela, ce rapport ne présente pas des résultats. La perte d'un employé essentiel à ce sujet contribuer à l'annulation de ce projet.

INDEX

1. **Introduction** 3

2. **Modifications to the Manufacturing Process** 4

3. **Image Reconstruction** 8
   - 3.1 Derivation of Equations 10
   - 3.2 Creating an Image by Summing Reflections 13
   - 3.3 The Problems of Reconstructing with Transient Data 14
   - 3.4 Interpolating Missing Data 15
   - 3.5 Estimating Fault Location 17
   - 3.6 Interpolating Image Plane Points 20

4. **The Program** 22
   - 4.1 Program Outline 22
   - 4.2 Housekeeping 24
   - 4.3 Data Acquisition 26
   - 4.4 Examining Time Histories 27
   - 4.5 Image Reconstruction 29

5. **Testing and Results** 31

6. **Project Review** 34

7. **Conclusions and Recommendations** 36

Appendix A - Transducer Design

Appendix B - Mechanical Drawings

Appendix C - Electronic Circuit Diagrams

1. Introduction

This report covers the second and final phase of the design and construction of a 2-D acoustic array for imaging defects in steel weldments. Phase I covered the design and construction of:

- the 100-element array,
- the transmit and receive electronics, and
- the interface for an IBM PC-AT compatible computer.

The design of the array and associated electronics was fairly straightforward, although keeping the physical size to a minimum was difficult. During Phase I, the only significant problem encountered was the manufacture of the array head. In particular, the sheet PZT crystal kept separating from the backing material when it was being cut into individual transducers. Although a working array was completed, it did not perform as well as desired.

This Phase II report covers:

- an improved manufacturing process,
- the mathematics of the image reconstruction procedure, and
- the software that was developed.

The new manufacturing process requires the PZT to be cut into individual elements before being attached to the backing. The image reconstruction procedure is based on the principles of geometric optics, phase shifts being added to digitized signals in software in the same manner that a physical lens phase shifts light. Other issues discussed include the difficulties of using transient data, interpolating data at transmitter element locations, and estimating the location of a defect. The section on software describes the program that was developed for data acquisition, displaying time histories, image reconstruction, and image display.

This report also covers the problems encountered when the array was tested with a steel test block. Useful data could not be obtained because the receiver gains were too low, consequently, no test results are presented. Although an improved receiver was designed and a prototype built, other factors prevented upgrading the array. One key staff person, responsible for the design of the new receiver (and the original array), left the subcontractor, Hugh Jones and Associates. As well, Fleet Technology decided it was not interested in further work on the array because of the financial risks involved and the heavy reliance on the performance of its subcontractors. Consequently, the project was not continued.

Potential problems of trying to use the array as originally built are also identified in this report. The problem of imaging defects very close to the array surface is seen as a major concern.

2. Modifications to the Manufacturing Process

In Phase 1, the array head of Fig. 1 did not perform as well as expected because of difficulties encountered in its manufacture. The original manufacturing process consisted of the following steps:

(1) One hundred syringe needles containing very fine #34 wires were arranged into a ten by ten grid.

(2) Tungsten-loaded epoxy was poured around the needles to hold them in place. The ends of the needles were visible on the side facing the circuit boards but not on the other; i.e., one end was buried in the epoxy. The wires came all the way through, however.

(3) After hardening, the side of the epoxy block with just the wires showing was ground smooth. Afterwards, the ends of the wires appeared as a grid of evenly-spaced dots.

(4) A sheet of PZT crystal was glued to the epoxy backing with an electrically conductive cement.

(5) The sheet was cut into a grid so that each square was centered above the end of a wire. The cuts were made slightly deeper than the thickness of the PZT and the conductive cement so that the elements would be electrically isolated from one another.

When the PZT crystal was cut into individual elements, it kept separating from the backing material. After several attempts, it became apparent that it would be necessary to individually re-glue those elements that fell off. Unfortunately, this resulted in some array surface irregularities and poor or nonexistent electrical contact between some elements and the wires embedded in the backing material. The deficiencies were felt to be serious enough to warrant manufacturing another array head before further work was performed. Because the problems were felt to be limited to the manufacturing process, they were not expected to be very difficult to solve.

During August, 1987, a very cooperative NRC agreed to assist in finding a suitable cutting procedure. A test piece was made consisting of a sheet of PZT glued to an epoxy backing as per the original design, but without the embedded wires. The NRC made a series of cuts in the PZT using a pulsed CO2 laser (75W avg. power, 170 hz), a high pressure water jet, and electric discharge machine (EDM). The spacing of the cuts was made to the design specification with a depth sufficient to electrically isolate the elements.

The EDM procedure did not cut the ceramic at all, in fact, it shattered the electrode. According to the technician that performed the work, he had not seen this happen before. The high pressure water jet cut the PZT but the material between a pair of parallel cuts was entirely lifted off of the backing. The laser at first appeared to work a little better as only some material near the edges was lifted. Unfortunately, under microscopic inspection there were clear signs of substantial cracks that ran parallel to the laser cuts from one side of the test piece all the way to the other.

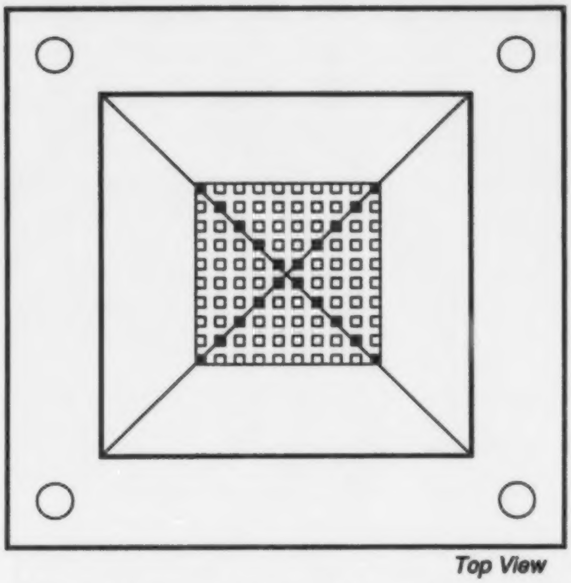
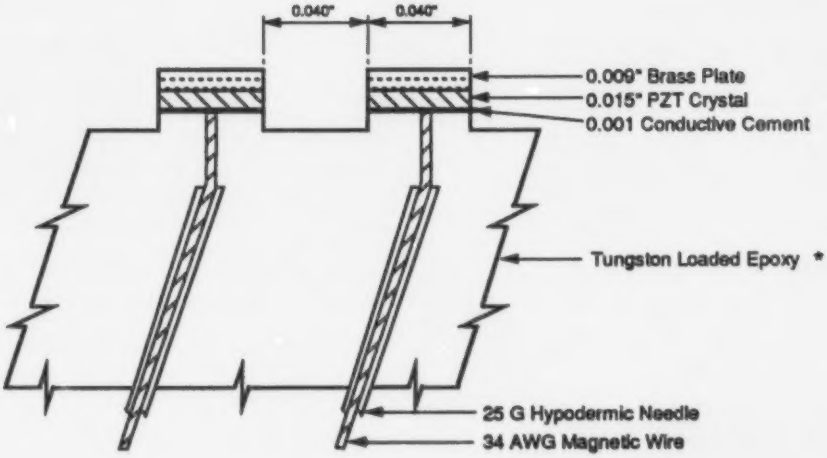


Figure 1(a) Configuration of the Transmitting Elements on the Phased Array


0.040"
0.040"
0.009" Brass Plate
0.015" PZT Crystal
0.001 Conductive Cement
Tungston Loaded Epoxy *
25 G Hypodermic Needle
34 AWG Magnetic Wire

* Component	Parts by Weight
Araldite 502	100
hardener HY956	20
Tungsten Powder	960

Figure 1(b) Detailed Structure of the Phased Array

By the time all these experiments were completed, MTC, the new welding research division of Fleet Technology, had moved into the Kanata facilities and a slow speed diamond wheel saw became available for use. This method did not work either but the source of the cutting problem became evident. When a test piece was being inspected sometime after being cut, not only did a piece of ceramic literally pop off into the air, but there was also epoxy cement on both the backing and the PZT. This indicated failure through the thickness of the cement itself, rather than either the cement-backing or cement-PZT bonds. This and the results of the cutting experiments indicated the presence of significant residual stresses due to the shrinkage of the cement. It became apparent that success hinged on finding a better cement, rather than finding a very gentle cutting procedure.

After some consulting with outside sources, H. Jones and Associates obtained a new conductive epoxy cement with supposedly better qualities. A new test piece was made but again no luck was had with the diamond saw. Consequently, an entirely new method was tried in which the PZT was cut into individual elements before it was glued to the backing. To position the elements, a grid form was manufactured with square holes just slightly larger than the PZT elements, but thinner than the thickness of the PZT. It was made from a dental material that would stick to the epoxy backing, but not permanently, thus, it could be removed afterwards as can paper glued to metal with rubber cement. After the form was glued onto the backing and the elements dropped into place, a flat weight was placed on top. Because of the differences in thickness, the weight only touched the PZT elements and not the grid form, thus a good mechanical bond and flat array surface was ensured. After the cement hardened, the grid form was peeled off.

Since the cement was conductive and the entire backing surface was covered, the elements still needed to be electrically isolated from one another. This was accomplished by cutting grooves into the backing through the thickness of the cement. Note, however, that the PZT elements did not have to be disturbed.

In summary, the revised manufacturing process is as follows:

(1) Prepare the desired transducer elements by cutting them from a sheet of PZT crystal.

(2) Arrange one hundred syringe needles containing very fine wires into a ten by ten grid.

(3) Pour tungsten-loaded epoxy around the needles to hold them in place. The ends of the needles should be visible on the circuit board side but not on the other; i.e., one end is buried in the epoxy. The wires need to come all the way through, however.

(4) After hardening, grind smooth the side of the epoxy block with just the wires showing. The ends of the wires should appear as a grid of evenly spaced dots.

(5) Put a layer of conductive cement on the backing and attach the grid form.

(6) Drop the PZT elements into place and place a weight on top of the assembly until the cement hardens.

(7) Remove the weight and peel away the form.

(8) If minor surface irregularities result due to varying cement thickness, grind down the elements using a mixture of oil and powder.

(9) Cut grooves into the backing material through the thickness of the cement using a slow speed saw or other suitable method.

The array head is now ready to be connected to the circuit boards.

The current array assembly was constructed using this procedure during the period December 1987-February 1988. The cutting of the PZT into individual elements was performed at Bell-Northern Research in Kanata, Ontario. The array head and electronics were mounted in a new aluminum housing that was a little larger than the one originally designed. It was able to accommodate additional circuit boards that were added at the end of Phase I.

3. *Image Reconstruction*

The image reconstruction procedure is based on the principles of geometric optics, using software to manipulate the digitized data as an idealized optical lens affects light. In a simple optical system, a single lens is used to add a phase delay to a wavefront passing through it, the amount of delay being inversely proportional to the radial distance from the center of the lens. See Fig. 2. The time delay is due to light's slower speed of propagation in glass than in air. The net effect is to curve the spreading wavefront so that it converges at some distance from the lens, creating an inverted image. The quality of the image is limited by the physical properties of the lens, such as its finite aperture, its thickness relative to the other distances involved, etc.

The acoustic image reconstruction procedure is based on the same principles with the array (combined with some software) acting as the lens. The acoustic signals at the array are digitized, producing eighty channels of time history information. A phase shift is added by using offsets into the vectors of data, the amount being a function of the desired magnification and focusing depth. An image may then be calculated by combining the data as if the acoustic energy had passed through the array and was focused using a physical lens.

Characteristics of the acoustic array system not shared with the optical analogy include:

(1) The acoustic signals used with the array are transient in nature, rather than continuous. This makes the image reconstruction procedure much more difficult.

(2) The acoustic data being manipulated is a sampled form of the original, both in time and in space. The spacial discretization results from using an array of receiving elements, while the temporal discretization results from digitizing the signals with a high speed analog/digital convertor.

(3) The errors associated with the sampling process become a concern. For instance, sampling rates and measurement word length need to be considered.

(4) Since very high speed samplers may only capture a limited amount of data at a time, the number of measurements that are available for the reconstruction process is limited. This also determines how far away a defect may be before its reflected signal falls outside the window of data that can be captured.

(5) By implementing the phase delays in software, some of the limitations of a physical lens are no longer a concern. For instance, the software "lens" is perfectly thin and could in theory introduce both phase delays and advances, something that would be impossible with a physical lens.

(6) The software "lens" can change its properties arbitrarily at any time to optimize its ability to focus on some particular point in space.

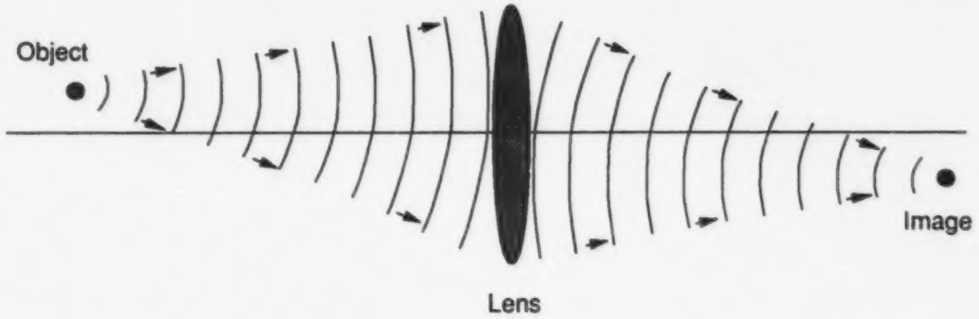

Object
Image
Lens

Figure 2 Simple Optical System

3.1 Derivation of Equations

The equations are derived without concern for the fact the acoustic energy is transient in nature. This factor is dealt with later. The following discussion assumes that material defects may be modelled as a collection of point sources, and that initially the depth of a defect below the surface of the material (or lens plane) is known.

Fig. 3 demonstrates the ideal case where there is a single point source P lying at coordinates (px, py, pz), pz being the depth within the material. To form an image of P, any rays leaving it should all arrive in phase at a corresponding point Q on the image plane. The location of Q at coordinates (qx, qy, qz) is determined by:

- *A straight line running between P and Q. It passes through the point O, the origin of the lens plane that is taken to be at the center of the array.*
- *The distance qz from the image plane to the lens plane. This distance is the product of pz and M, a user-specified magnification factor.*

These relationships govern all of the other equations. The phase shift that the lens must provide may now be determined.

Consider a ray leaving P, hitting the lens plane at an arbitrary point R, and travelling on to Q. If R is located at (rx, ry, rz) and Rz = 0, the total distance is given by:

$$d1 = \sqrt{(rx-px)^2 + (ry-py)^2 + pz^2} + \sqrt{(rx-qx)^2 + (ry-qy)^2 + qz^2}$$

If R was located at the origin (point O), d1 would be the shortest distance possible between P and Q. In this case, the distance is given by:

$$d2 = \sqrt{px^2 + py^2 + pz^2} + \sqrt{qx^2 + qy^2 + qz^2}$$

For the energy travelling along an arbitrary path to arrive in phase with that travelling along the shortest path, the lens must introduce a phase shift, ϕ, to make the distances equal; i.e. the lens must enforce the relationship:

$$d1 + \phi = d2$$

The lens may either delay those rays passing closer to the origin, or advance those that are further away. A phase advance was chosen so that a ray passing from P to Q through the origin O has zero delay. Consequently, ϕ is a negative value given by:

$$\phi = \sqrt{px^2 + py^2 + pz^2} + \sqrt{qx^2 + qy^2 + qz^2}$$
$$- \sqrt{(rx-px)^2 + (ry-py)^2 + pz^2} - \sqrt{(rx-qx)^2 + (ry-qy)^2 + qz^2}$$

This equation may be simplified using the relationships:

$$qx = -Mxpx$$
$$qy = -Mxpy$$
$$qz = -Mxpz$$

where px, py, and pz are the coordinates of point P. Thus the phase shift added by the lens is given by:

$$\begin{aligned}
\phi = {} & \sqrt{px^2 + py^2 + pz^2} + \sqrt{(-Mxpx)^2 + (-Mxpy)^2 + (-Mxpz)^2} \\
& - \sqrt{(rx-px)^2 + (ry-py)^2 + pz^2} - \sqrt{(rx-(Mxpx))^2 + (ry-(Mxpy))^2 + (-Mxpz)^2} \\
= {} & \sqrt{px^2 + py^2 + pz^2} + M x \sqrt{px^2 + py^2 + pz^2} \\
& - \sqrt{(rx-px)^2 + (ry-py)^2 + pz^2} - \sqrt{(rx + Mxpx)^2 + (ry + Mxpy)^2 + (Mxpz)^2} \\
= {} & (1+M) x \sqrt{px^2 + py^2 + pz^2} - \sqrt{(rx-px)^2 + (ry-py)^2 + pz^2} \\
& - \sqrt{(rx + Mxpx)^2 + (ry + Mxpy)^2 + (Mxpz)^2}
\end{aligned}$$

Since the phase shift is a function of the coordinates of P, no fixed lens can be perfect for an arbitrary source location. This relationship is usually simplified for basic optical systems by assuming that the distances rx-px, ry-py, rx+Mxpx, and ry+Mxpy are negligible compared to pz or Mxpz. This is a good approximation provided the size of the object is small compared to its distance from the lens.

The exact formula for ϕ was implemented in the software because it was very easy to do so. Consequently, the lens properties are optimized for each point being calculated on the image plane. The use of a simplified expression would have reduced processing times at the expense of accuracy.

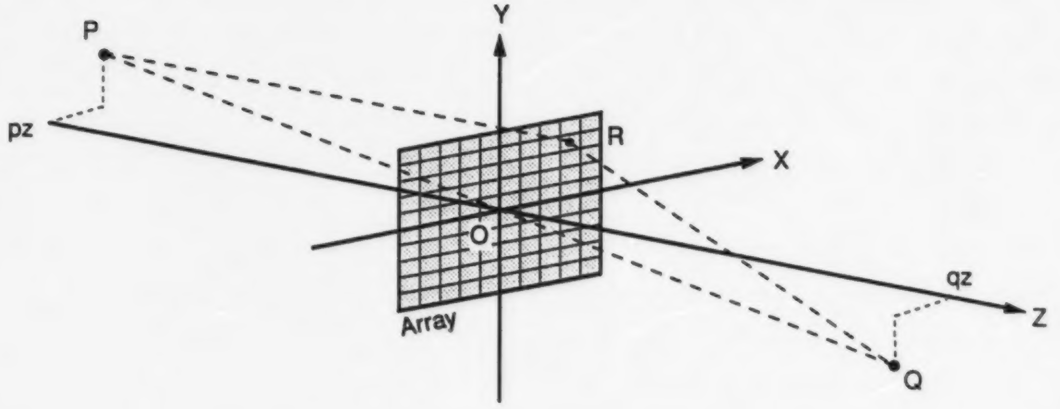

Y
P
pz
R
X
O
Array
qz
Z
Q

Figure 3 Acoustic System Coordinates

3.2 Creating an Image by Summing Reflections

An image is constructed by calculating a 2-D matrix of values representing a sum of the acoustic wave intensities as they "pass" through a point on the image plane, and then representing these values on a monitor by coloured squares. Assume for the moment that the intensity at a point Q on the image plane is to be calculated. If there is a single defect at a point P corresponding to Q, all of the waves that travel from P to the lens and on to Q will be in phase once their phase differences have been corrected by the lens. If there is a defect but it is at a point not matching that of Q, the rays will tend to cancel each other because they will not arrive exactly in phase (unless aliasing occurs). Thus, the greatest intensity values indicate where the defects are located. When these values are represented with coloured squares, the locations of defects are revealed visually as if an image of the interior of the material was being displayed.

Each intensity value is calculated from a double sum. The first totals the instantaneous values of the acoustic signals (travelling along the rays) as they pass through some point on the image plane. The second sums the squares of these total values, integrating them in time in the same way that photographic film integrates the average light intensity to which it is exposed. If the total instantaneous values were not squared before being summed, the sum of the values for in-phase rays would be zero if taken over an integer number of cycles. This would not produce the desired result.

A simplified form of the algorithm for image reconstruction is as follows:

```
1.      Input the desired inspection depth, the magnification, and the number of
        time steps to be used.

2.      For each point Q on the image plane:

        2.1     For the specified number of time steps:
                2.1.1 For each element E on the array surface:
                        2.1.1.1     Calculate the distance d from E to Q.
                        2.1.1.2     Calculate the lens phase shift, $\phi$.
                        2.1.1.3     Determine the value of the signal as it reaches
                                    Q from E, given d and $\phi$.
                        2.1.1.4     Sum the instantaneous value with those from
                                    the other elements on the array surface.
                2.1.2 Square the sum of the instantaneous values.
                2.1.3 Sum the squared value with those from previous
                      time steps.
        2.2     Determine a colour corresponding to the sum of the squares.
        2.3     On the screen, draw a coloured box at point Q on the screen, taking
                into account image inversion.
```

Steps 1 and 2.1 indicate that only data over a certain number of time steps is used. Section 3.3 explains why.

3.3 The Problems of Reconstructing with Transient Data

Performing image reconstruction on a desktop computer results in some data processing problems. Not only is there a very large amount of data (up to 400,000 measurements from 100 measured and interpolated channels), the nested loops of the algorithm can require a huge amount of calculation. To aggravate this further, the MS-DOS operating system makes accessing such large amounts of data very awkward; i.e. programs are essentially restricted to 640 kilobytes of code and data. While the use of an expanded memory board "breaks the 640K barrier" so that several megabytes of memory become available, it imposes other problems. A program can only access a total of 64 Kbytes of expanded memory at a time through four 16K windows. This requires a program to keep track of what data is currently visible and to move the 16K windows as required. Since only four channels of data will fit into one window at a time and moving windows can be time consuming, some compromises were felt to be necessary.

The solution was to limit the amount of data to be used in the reconstruction procedure; i.e. rather than integrate in time over all of the available data, the software was designed to integrate the data over a limited number of time steps. Since the data is transient in nature, virtually all of the useful information would be contained in just a few cycles of data anyway. Such an approach should also improve the signal to noise ratio.

A drawback to this approach is analogous to the difficulty of using a flash with a camera. The fact that the light from the flash is only present for a very short time requires that the normally closed shutter be opened at just the right moment to let the light in. In the acoustic system, the signal reflected from a defect may only be present for a few cycles, and the number of time steps used in the reconstruction procedure could be on the same order. Thus, it is very important that the most significant data be captured.

To select the data for the image reconstruction, the user must specify:

- a time offset to indicate the first measurements to be used, and
- the number of time steps over which the data is to be integrated.

The first value is used to eliminate data preceding any reflections of interest, while the second indicates how many measurements are to be used from each time history. Once these have been specified, the data is read into main memory from the expanded memory card. This allows it to be accessed much more quickly.

3.4 Interpolating Missing Data

Since all of the elements along the main diagonals of the array are transmitters, no measured data is available at these locations. Since this lack of data can only have adverse affects on the quality of the image reconstruction, ways of interpolating data from surrounding channels were investigated. Four methods were developed, each tested using simulated data for the cases of one and two point sources located at various distances from the array. Visual comparisons between the interpolated and theoretical data indicated that three of the methods had at best satisfactory performance, and that one was not usable at all.

Method 1:

Fig. 4(a) demonstrates the simplest method. At each time step, the missing value is calculated from the average of the four nearest values. When data is being calculated for a corner element or for an element at the center of the array, a missing value is calculated by linearly extrapolating from the two nearest elements in perpendicular direction, and taking the average of the two.

Method 2:

The second method is similar to the first except it uses a weighted average in time of the values calculated using Method 1. If the values $u(k-1)$, $u(k)$ *and* $u(k+1)$ *are the values estimated using Method 1 at time steps k-1, and k+1 respectively, Method 2 estimates the missing value* $V(k)$ *from*

$$v(k) = (u(k-1) + 2*u(k) + u(k+1)) / 4$$

Note that twice as much emphasis is placed on the value for the current time step.

Method 3:

The third method is similar to the first in that it uses only data from the current time step. It uses cubic, rather than linear, interpolation, however. To achieve this, data is used from twice as many elements as indicated in Fig. 4(b). This method was the slowest of the three, but it produced the best results of all four methods.

Method 4:

This method was based on a discretized form of the wave equation. It appeared to be unstable, presumably due to the relatively large spatial step size (determined by the element spacing) compared to the very small time step (resulting from a 100 MHz digitizing rate). Its results were totally unusable.

Because the methods were compared with only a couple of sets of data and without the benefit of accurate measurements, there is no guarantee that Method 3 is the best. As well, other untested approaches could be better.

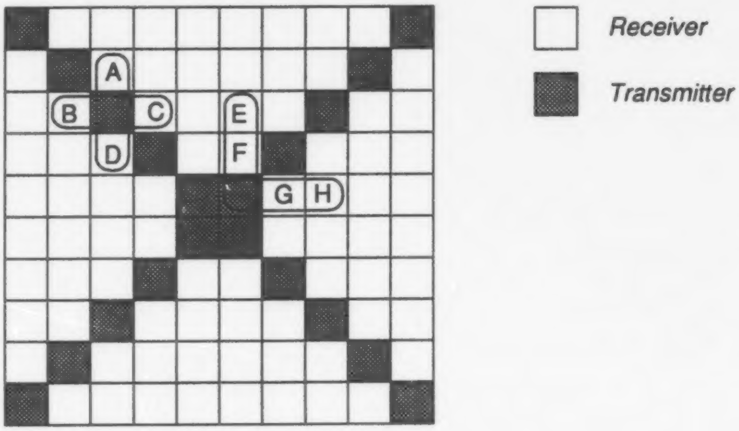

A
B
C
D
E
F
G
H
Receiver
Transmitter

Figure 4(a) Interpolating Using Method I

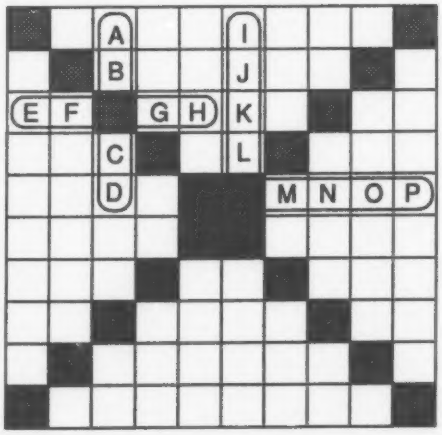

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P

Figure 4(b) Interpolating Using Method III

3.5 Estimating Fault Location

Each image reconstruction depends on several parameters such as inspection depth and magnification. Performing an exhaustive search is not practical with a desktop computer, thus a means was sought that would allow the location of a defect to be estimated from the time of flight information. The location could then be used to provide input parameters for the image reconstruction. Fig. 5 demonstrates one approach with a 2-D example. It uses a single transmitter, Tx, and two receivers, Rx1 and Rx2. The times of the reflections, t1 and t2, indicate the path lengths from the transmitter to the defect, D, and back to the receivers. Assuming the positions of the transducers are known, the location of D can be determined from the intersection of the two ellipses that correspond to the path lengths shown in Fig. 6.

Unfortunately the equations for the 3-D case become very complex, particularly when there are multiple transmit elements and the transmitters are not all fired at the same time (as provided for in the array design to provide a steerable beam). Since only an approximate solution was sought, another approach was used.

The problem was formulated using an optimization technique that tries to find the location of a defect that minimizes the sum of the errors between the measured time to peaks and the theoretical time to peaks that would occur if there was a single point defect. Fig. 7 illustrates the approach for a 2-D problem. Let L0 be the path length from a defect at (xd,yd) to the transmit element that results in the earliest reflected signal. (This will be the transmitter closest to the defect only if they are all fired at the same time.) Li is taken to be the distance from the defect to receiving element i, and Tpi is the time to the reflection measured at element i. The location of the defect is estimated by adjusting its position to minimize the sum of the squared errors between the measured times to peaks, Tpi, and the times resulting from the path lengths L0 plus Li; i.e. the values of xd, yd and zd are adjusted to minimize:

$$Z = \sum_{i=1}^{N} \left(\frac{L0 + Li}{C} - T_{pi} \right)^2$$

where N = the number of data channels to be used

c = the speed of propagation

$L0$ = the distance between (xd,yd,zd) and the transmitter that results in the earliest reflection, and

Li = $((xd-xRxi)^2 + (yd-yRxi)^2 + zd^2))^{0.5}$

= the distance from the defect at (xd,yd,zd) to receiver i at (xRxi, yRxi, 0)

This method can use data from as many channels of data as desired. The version implemented in the software uses data from four receivers that are nearest the corners of the array.

The software uses a Hooke and Jeeves optimization procedure to minimize the objective function. It also assumes that all transmitters are fired at the same time, even though this will not be true if the transmit beam is steered off axis.

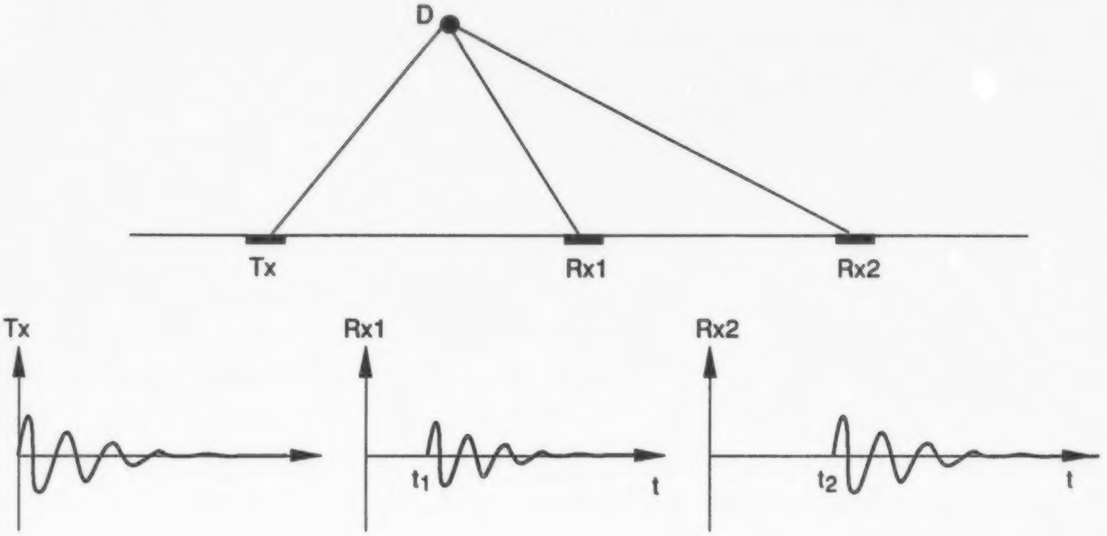
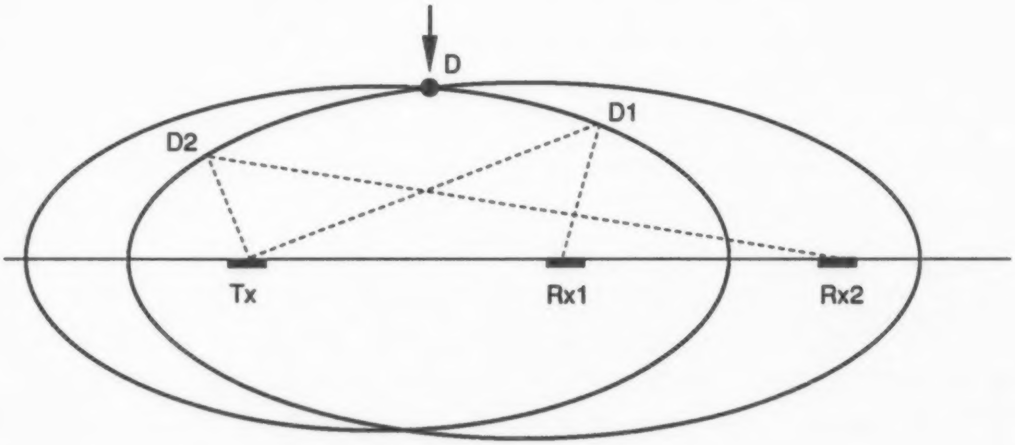

D
Tx
Rx1
Rx2
Tx
Rx1
Rx2
t_1
t
t_2
t

Figure 5 Transducers and Time Histories


D
D1
D2
Tx
Rx1
Rx2

$$\overline{TxD1} + \overline{D1Rx1} = t_1 \times c$$
$$\overline{TxD2} + \overline{D2Rx1} = t_2 \times c$$

Figure 6 Determining Location of D using Ellipses

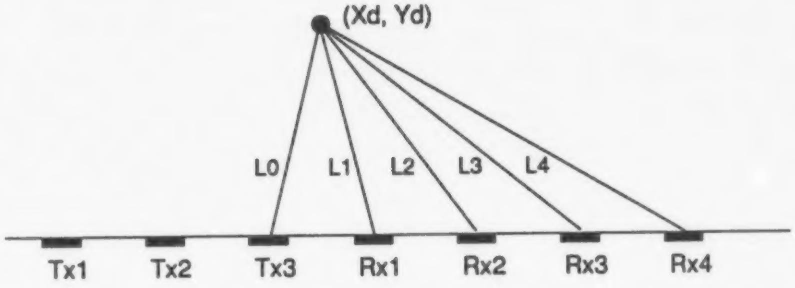

(Xd, Yd)
L0
L1
L2
L3
L4
Tx1
Tx2
Tx3
Rx1
Rx2
Rx3
Rx4

Figure 7 **Estimating Location of D using Optimization Technique**

3.6 Interpolating Image Plane Points

Calculating points on the image plane is a relatively time-consuming effort, and so it best to calculate as few as possible. On the other hand, if too few are calculated, the image may be too coarse to reveal significant details. Consequently, a scheme was developed so that image plane points could be interpolated from those that were calculated directly.

The procedure is based on Shannon-Whittaker interpolation. When used for one dimensional problems, it may have zero error provided the sample points are spaced sufficiently close together. The value f(x) is calculated from the sum of N weighted sinc functions as follows:

$$f(x) = \sum_{i=1}^{N} a_i \sin\left(\frac{x - x_i}{dx}\pi\right) \Bigg/ x - x_i$$

The a_i values are the values of f(x) on an evenly spaced grid of N data points separated by distance dx. See Fig. 9. If x equals some particular x_i value, say x_j, the argument of the sine function will be an integer multiple of pi for all i since the distance $x_j - x_i$ will be a multiple of dx. Thus, all terms will disappear except for $x_i = x_j$ (because sin(0)/0 equals one). Thus, $f(x_i) = a_i$; i.e. the interpolated curve will always pass through the measured a_i values, and the closer x is to a measurement point x_i, the greater the contribution of a_i to f(x).

A two dimensional version was implemented in which the interpolated values are given by:

$$f(x,y) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \sin\left(\frac{\pi r}{dr}\right) \Bigg/ r$$

where $dr = dx = dy$

$$r = \sqrt{(x - x_i)^2 + (y - y_j)^2}$$

The software allows an image to be shown with and without the interpolated data present.

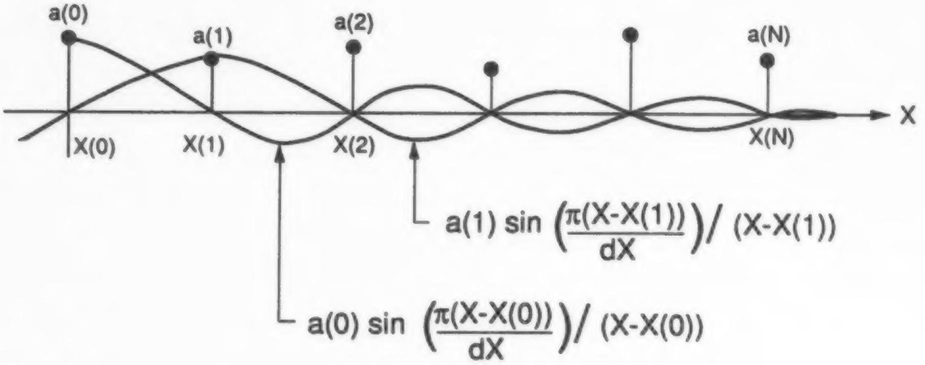

a(0)
a(1)
a(2)
a(N)
X
X(0)
X(1)
X(2)
X(N)
a(1) sin $\left(\frac{\pi(X-X(1))}{dX}\right)/\ (X-X(1))$
a(0) sin $\left(\frac{\pi(X-X(0))}{dX}\right)/\ (X-X(0))$

Figure 8 Shannon-Whittaker Interpolation
dX = X(i)-X(i-1) for i = 1 to N

4. The Program

4.1 Program Outline

The program for performing the image reconstruction, named AC, was developed in Microsoft C V4.0 on a Compaq 286 computer. It should run on all IBM PC-AT compatibles equipped with EGA graphics and a 1 Megabyte expanded memory board such as an Intel AboveBoard. The user interface is that of a command line interpreter (CLI) combined with function key assignments. A CLI selects program options by decoding character strings entered from the keyboard. This is the standard interface on large time-shared computers. With the aid of an on line Help facility, the program is very easy to use.

To run the program, enter AC at the DOS level and the user prompt "->" will appear. All options may be chosen by entering the name followed by the Enter key. If an option has been assigned to a function key as indicated in the Table 1, it may be selected by simply pressing the key. Prompts will be given for any necessary arguments.

The options of the program may be roughly grouped into four groups. The first includes housekeeping utilities such as opening files; commands in the second group are used for data acquisition; those in the third are used to examine time histories of measured data; the fourth group covers image reconstruction procedures. Before the details of specific commands are discussed, their format should be explained:

- Keywords are given in uppercase letters to distinguish them from other text, even though the CLI is not case sensitive. For instance, entering "PLOT" or "plot" has the same response.

- Any letters appearing within square brackets may be omitted.

- Italicized lowercase letters represent other text to be specified by the user.

- In general, if required arguments are not included with the option keyword (always the case if function keys are used), prompts will be issued for them.

- If the wrong command has been issued, entering the keyword CANCEL will prevent any values from being changed.

Table 1

Function Key Assignments

FUNCTION KEY	OPTION
Esc	Quit
F1	Help
F2	Open
F3	Close
F4	List Files
F5	Clear

4.2 Housekeeping

The procedures in this category include keywords for terminating the program, using the help facility, accessing files, and running DOS commands.

QUIT:

To terminate the program and return to the DOS command interpreter, enter QUIT. Any open files will be closed automatically.

HELP:

HELP is used to access information on program keywords when their names or use have been forgotten. If HELP alone is entered, a list of the program keywords is given. Entering:

HELP x - provides information on keyword x.

SET:

SET is used to specify various parameters for plotting time histories, acquiring data, reconstructing images, etc. The format is:

SET parameter-name value

where the arguments give the name and new value of the parameter. The old and new values are echoed for verification. If the value is not given, the user is prompted to enter one. (CANCEL may be entered to abort changing the parameter.) If the name of the parameter is not specified, the user is told to use SET with one from a given list.

OPEN:

A file of data may be opened by entering:

OPEN filename

where filename is the name of the file as it appears in the DOS directory. Quotes should not be used, and any file extension should be included, as in:

OPEN TEST_1.DAT

Only one file may be open at a time. Trying to open more than one will result in an error message. If the file exists, any data it contains will automatically be read in, overwriting any data currently in memory. If the file does not exist, a new one will be created that can be used to store data using the WRITE command.

WRITE:

Data may be stored using:

WRITE

This will write any test results (acquired using the SCAN command) to the currently open file. The file must have been opened previously (with the OPEN command), otherwise an error message is given.

CLOSE:

A file may be closed using:

```
CLOSE [filename]
```

with or without the filename specified. If one is currently open, its name will be echoed when it is closed. If filename is specified and it is not the name of the currently open file, an error message is given.

LIST:

LIST requires one of two arguments, namely FILE[S] or PARAM[ETERS]. To see which file of measured data is open, enter:

```
LIST FILE[S]
```

Entering:

```
LIST PARAM[ETERS]
```

will result in the listing of all of the data acquisition and image reconstruction parameters. If LIST is entered with no argument specified, the user will be asked if files or parameters should be listed. Acceptable responses are FILE[S], PARAM[ETERS], or CANCEL.

TIME:

TIME causes the current time to be printed.

Other commands:

Other programs and DOS functions can be invoked from within AC by calling them as if the user was at the DOS level. For instance, entering:

```
DIR
```

will get a directory listing. However, this capability must be used with caution, particularly when another program is run. If it is a simple program, you should be able to return to AC when it terminates. Unfortunately, if this other program accesses expanded memory, data intended for AC may be lost. Another potential problem is it could change the graphics card settings from those initialized by AC. If possible, this capability should only be used to access simple DOS commands.

4.3 Data Acquisition

Parameters for SET:

The two parameters that affect data acquisition are listed below. Each may be changed using SET followed by the parameter name and a value.

PHI: *PHI is the angle that the transmit beam may make with respect to a vector that is normal to the plane of the array. Its range is approximately 48.6 degrees in 2.8 degree increments.*

THETA: *THETA is the azimuth angle of the transmit beam. It must be a multiple of 45 degrees (because all of the transmitters are on the main diagonals of the array).*

Values for PHI and THETA are rounded off whenever they do not correspond to the settings that may be represented in the hardware.

SCAN:

SCAN performs two separate procedures. The digitizer and array electronics are initialized, and then the transmit-receive loop is executed eighty times, once for each receiver channel. If SCAN is used, any other data currently in memory will be overwritten, regardless of whether it was obtained from a previous use of SCAN or it was read into memory from a file.

4.4 Examining Time Histories

PLOT:

Each time history of measured (or interpolated) data may displayed by specifying:

```
PLOT channel-number
```

where channel-number is between 1 to 100. If a transmit channel is indicated before data has been interpolated for it, only zero's will appear. If no labels or plot parameters have been specified (discussed under ***Parameters for SET****), both the abscissa and ordinate will be autoscaled. After reviewing the plot, pressing any key returns the user to AC's command line interface.*

PLOT may be called over and over with different channel numbers. Each time it is used, the new data will be superimposed on the previous plot. Clipping will occur if the new data exceeds the existing plot scales.

Parameters for SET:

The following lists the plotting and interpolation parameters that may be specified using the SET command:

TITLE1: The title that appears at the top of the plot.
TITLE2: The title that appears just below TITLE1.
TITLEX: The label for the abscissa.
TITLEY: The label for the ordinate.
XMIN, XMAX, and XTICK: These affect the scaling of the plot's abscissa. The user may specify the minimum and maximum values by setting XMIN and XMAX as desired. If XMAX is set equal to or less than XMIN, the abscissa will be autoscaled. XTICK controls the size of the tick mark increments.
YMIN, YMAX, and YTICK: These affect the plot's ordinate scaling in the same way that XMIN, XMAX, and XTICK affect its abscissa.
INTERP_METHOD: Selects the interpolation method to be used. Only values of 1, 2 or 3 may be used. These correspond to the methods described in Sect. 3.4.
INTERP_CH: INTERP_CH specifies the channel number for which a time history of data may be generated. Since all of the transmitters are on the diagonals of the array, only the following channel numbers should be used:

1, 12, 23, 34, 45, 56, 67, 78, 89, 100 or
10, 19, 28, 37, 46, 55, 64, 73, 82, or 91.

Examples of the use of SET are:

```
SET TITLE1 First Test Results
SET XMIN 0.05
SET INTERP_CH 12
```

INIT_PLOT:

INIT_PLOT clears the graphics screen of old plots, resetting the plot parameters that may have been set previously. For instance, XMIN is set equal to XMAX so that the next time PLOT is used, the graph will be autoscaled. INIT_PLOT does not affect the titles, however.

CLEAR_PLOT:

CLEAR_PLOT behaves the same as INIT_PLOT with the exception that the graph scales are not reset, thus the current plot scales will be used the next time PLOT is used.

INTERP_CH:

This command is used to create one channel of interpolated data for a transmit element location. The channel number and the method used to perform the interpolation are specified using the SET command.

INTERP_ALL:

INTERP_ALL interpolates data for all of the transmit element locations using the currently specified interpolation method.

FIND_SIGNIF:

It is often convenient to have the plots show only the most significant data on an expanded scale, rather than leaving it up to the user to set appropriate xmin and xmax values. Since the raw data is measured on a scale of -128 to +127, FIND_SIGNIF searches an indicated channel of data for the first point, Xth, that is above a threshold of 10. It then sets the first data point to be plotted to the one 30 measurements before Xth, and the last to 275 points after the first. Assuming the sampling frequency is twenty times the fundamental frequency of the acoustic signal, the plot will show approximately 12 cycles of data.

FIND_SIGNIF should be used in a command sequence that will benefit from its results, as in the following:

```
INIT_PLOT            (to reset all of the plot parameters)
FIND_SIGNIF channel-number (to find the significant data)
PLOT channel-number       (to see the data on an expanded scale)
```

The call to INIT_PLOT is used to ensure that autoscaling will occur for both the abscissa and ordinate. If CLEAR_PLOT had been used instead, the previous plot's scales would remain in effect and they might not be appropriate.

4.5 Image Reconstruction

Parameters for SET:

The parameters that affect the image reconstruction are listed below. Each may be changed using the SET command followed by the parameter name and a value.

MAGNIFY:	*Specifies the magnification factor. If set to unity, the displayed image corresponds to the size of the array head.*
DEPTH:	*Specifies the inspection depth for which an image will be created.*
INT_STEPS:	*Specifies the number of time steps over which the measured data is to be integrated.*
START_TIME:	*Specifies the time offset to the first data points to be used in the image reconstruction process.*

ESTFAULT:

ESTFAULT estimates the location of a point defect that would correspond to the peak reflections measured near the four corners of the array. ESTFAULT prompts the user for a starting location for the optimization procedure.

RECON:

RECON executes the image reconstruction process. It does not display the image, however; IMAGE or INTERP_IMAGE must be used to show it.

INTERP_RECON:

INTERP_RECON interpolates points on the image plane to double the resolution of the displayed image. RECON must be used first to generate the data from which the new values will be calculated. The interpolation procedure is described in Sect. 3.6.

IMAGE:

IMAGE is used to display the image constructed using RECON. First, the program will enter the graphics mode and a scale that maps image intensities to colours will be presented. Entering "A" will autoscale the list of colours so that each represents an equal range of intensities. The ranges may also be changed individually; first the colour number is entered and then prompts will appear so that the minimum and maximum values may be specified.

Every time that the scale is changed, the image is redrawn as a matrix of coloured squares 10 by 10. Each value from the matrix of image intensities is compared to the range each colour represents, starting with the colour at the bottom of the screen (black). When the intensity is found to fall within a range, the corresponding colour is used. The process of starting with the colour at the bottom of the screen also controls how the image will appear when the ranges of some colours overlap, i.e. the colour nearest the bottom of the list will be used. The ranges may overlap if the ranges are changed individually;

Entering Q exits the graphics mode and returns the user to the CLI prompt.

INTERP_IMAGE:

INTERP_IMAGE displays the image created using INTERP_RECON; i.e. the image is drawn as a 20 by 20 matrix of values rather than as a 10 by 10 matrix. INTERP_IMAGE is used in exactly the same manner as IMAGE.

5. Testing and Results

After the software was developed, arrangements were made to ship the array and other test equipment from H. Jones and Associates to the Kanata facilities. The test equipment included:

1. a 90V power supply (for driving the transmitter elements),
2. a 30 dB RF amplifier (for amplifying the signals from the receiver elements),
3. a 100 Mhz Lecroy digitizing card for use in a CAMAC crate (chassis),
4. a Lecroy GPIB interface card for a CAMAC crate.

Because of previous commitments, H. Jones and Associates were unable to supply their CAMAC crate as originally planned. Consequently, an identical piece of equipment was rented from Rayonics in Ottawa. A 15 volt dc power supply was supplied by Fleet Technology for powering the RF amplifier.

The equipment was connected as shown in Fig. 9. Immediately, there was a problem with the digitizer and the computer that appeared to be software related. After having scanned a channel of data, the Lecroy GPIB interface would immediately start transmitting the data to the computer even though it had not been sent the command to do so. Eventually, a broken wire was found inside the Lecroy interface card, apparently a result of shipping. After the connection was repaired, the data acquisition software functioned properly.

The next stage was to test a 15.24 by 15.24 by 5.08 cm (6 by 6 by 2 inch) steel block provided by CANMET. It has a series of holes drilled on one face and one edge to act as defects. The array was mounted on the face opposite the holes, firing through the thickness of the block. The thickness of the block was thin enough that a reflected signal would fall within the window of data that the digitizer could capture. The data buffer in the digitizer allowed 4096 measurements to be taken at a rate of 100 MHz, resulting in a 40.96 microsecond window. Assuming a speed of propagation of 5000 m/sec, the thickest piece that could be tested was:

5000 m/sec * 40.96 microsec /2
= 10.2 cm

which is about double the thickness of the block, but less than its height.

Unfortunately, when the block was tested, no echo could be found in any channel of data. The digitizer used an 8-bit A-D convertor, thus a discretized signal could range from -128 to +127 in value. The levels seen with the test block were at best plus or minus 5 counts with no distinct echo from the other side; i.e. if any reflection was present, it could not be distinguished because the signals appeared as coarsely discretized noise.

To verify that the array was receiving the proper signals from the computer, an oscilloscope was used to monitor the reception of the trigger signal at the array, and to monitor the selected receiving channel. The presence of the trigger signal was verified, thus the array was being told to fire. Although it was not possible to determine voltage levels right at the transmit elements because of space restrictions, a weak pulse could be seen at

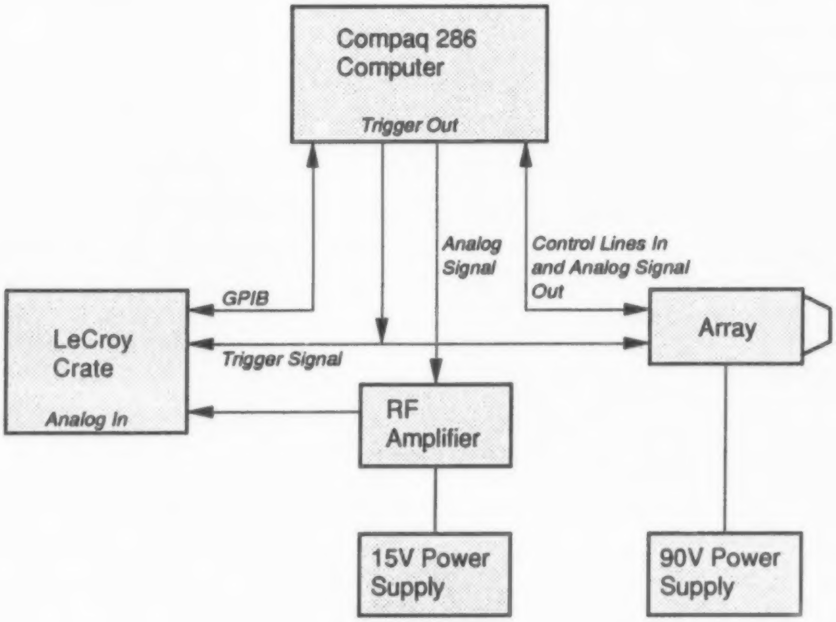

Compaq 286
Computer
Trigger Out
Analog
Signal
Control Lines In
and Analog Signal
Out
GPIB
LeCroy
Crate
Analog In
Trigger Signal
Array
RF
Amplifier
15V Power
Supply
90V Power
Supply

Figure 9 Test Equipment Connections

time zero in the scope trace monitoring the receiver. Its presence indicated that the transmitters were firing and the selected receiver was capable of detecting a signal. The weakness of the signal was not surprising because the receivers are only very sensitive to signals impinging at an angle normal to the plane of the array.

*The scope traces confirmed that the signals were too low to be properly digitized by the available equipment. The measured signals were less than 1 millivolt (peak), thus even with a 30 dB amplifier providing a voltage gain of 31.6, the signals would only be about 30 mV. The digitizer used an 8-bit A-D convertor with a full-scale window of 1 Volt, thus its sensitivity was approximately 4 mV per bit. Thus, a signal in the 0.8 mV range would only account for a change of 0.8*31.6/4 or 6 bits. This was close to the 5 bits that were measured and about twenty times less than the full-scale window of the digitizer.*

To try and improve the signal, another amplifier was borrowed from H. Jones and Associates; it was placed in series with the first amplifier. Although the levels improved, the signals still appeared as noise. Various low pass filter combinations were used to try and remove the noise, but no useful signals could be found.

At this point it became apparent that the array would have to be made more sensitive before useful tests could be made. Estimates for improved receiver amplifier boards were obtained from H. Jones and Associates, and CANMET was consulted. Fleet Technology eventually decided that it was best not to pursue the project any further because of projected losses. This decision was reinforced when it was learned that Dr. Kwan at H. Jones and Associates had moved to the United States. Dr. Kwan had been one of the key personnel involved in the array's design and construction.

6. Project Review

Both Phase I and Phase II of the project suffered from serious problems. Even though most of the key personnel had considerable experience with similar equipment, the major problems were not foreseen because an acoustic array of the type specified had not been built before. Such is the risk in any research-oriented project.

The purpose of the project was hampered in Phase I by manufacturing problems. This was due in part to the long chain of companies involved in the project. Fleet Technology, at the time Arctec Canada, subcontracted the design and construction to H. Jones and Associates. They, in turn, subcontracted Orion Electronics to build the device and gave them some leeway in the construction, allowing them to substitute components as necessary to ensure that the array was built in a reasonable length of time. This was a concern because the subminiature components required were difficult to obtain. Rigiflex of Halifax, the company Orion subcontracted to make the circuit boards, went out of business, thus Orion attempted to proceed with the boards that had been delivered. Orion made some circuit modifications that resulted in a loss of gain in the receiver section, changes that eventually contributed to problems in Phase II. Orion also had a great deal of trouble with the PZT crystal separating from the backing material when it was being cut into individual elements. They eventually resorted to individually re-gluing elements that came off during the cutting process.

When it was tested, the array did not perform as well as it should have. One of its problems, possibly due to the original design, was insufficient drive to the output transistors. This resulted in transmit signals that were too low to penetrate a four inch test block. In an attempt to improve its performance, H. Jones and Associates added two circuit boards that boosted the transmitter outputs. As discussed in the Phase I report, the changes were not entirely satisfactory because the transmitter outputs varied from 40 to 80 volts. In addition, the array head needed further work because many of the crystals made poor or no contact with the wires embedded in the backing. Although preliminary tests indicated that the array did work, it was not until the start of Phase II that the severity of the problems became apparent.

Fleet Technology spent a considerable amount of time and effort trying to manufacture an acceptable array head; i.e. finding a way to get the PZT elements to stay attached to the backing. As discussed in Section 2, a solution was eventually found and another head was constructed by H. Jones and Associates. By this time, however, each company involved in the project was spending time attempting to correct the problems that were rightly the responsibility of its subcontractor. When the problems of insufficient signal strength were encountered by Fleet Technology, it became apparent that the subcontractors could not guarantee a working system and were unprepared to attempt this without additional funding. At this point, Fleet Technology decided that it could not perform further work because of projected losses and the poor prospects of achieving a working system. It chose to accept responsibility for its chain of subcontractors instead.

Fleet Technology thought the prospects of getting a working system were poor for three reasons, even if good results could have been obtained with the test block. These were:

(1) It would only be possible to look for defects that were on the order of several inches away from the array surface, implying only very thick material could be tested. If a defect was much closer than that, the directivity characteristics of the transmitters would adversely affect the illuminating signal, while those of the receivers would prevent reflected signals from being measured properly.

(2) It would be very difficult to make good contact with the material under test because of the geometries involved. The array is flat, but the welded joints of tubular members would be complex curves. One idea was to try to use a steel block as a spacer, one end matching the shape of the member joint, the other end flat for mounting the array. Such a spacer would also help to solve problem (1) if the defects were near the surface. Unfortunately, the joint curvature of tubular members continuously changes, thus no single spacer would do.

(3) Another problem was the complication of surface roughness. While some surface irregularities could be compensated for using a layer of grease, the roughness of actual members would be difficult to accommodate.

Since the members used in offshore structures are usually tubular, less than several inches thick and can have quite rough surfaces, it did not appear that the array could ever be used for its intended purpose.

7. Conclusions and Recommendations

Although it was not possible to perform any tests with the acoustic array, much was learned about the process of manufacturing one.

- The basic electronic design for the control circuitry, transmitters and receivers is still felt to be acceptable although there is room for improvement. (See the recommendations below.) Considering how often new components become available, it is quite possible that new devices could provide better performance and reduced physical size.

- A reliable procedure for constructing the array head was developed, and the problems of using several other approaches was investigated. The only successful method involved cutting the PZT crystal into individual elements before being glued into place.

- A procedure was developed for estimating a defect location from the array time histories. Unfortunately, it was not possible to test it with actual data.

- The data processing problems of using a relatively small computer were investigated. It was found that image reconstruction can be performed, but it is not practical to use all of the available data.

- The beam steering characteristics of the array were investigated in Phase I. The results indicated that a very narrow beam was not achievable using so few transmit elements in pulsed mode.

If further work was to be put into the array, the following is recommended:

- The receiver gain needs to be improved by at least a factor of five.

- The transmitter electronics needs to be modified so that the outputs do not vary as much.

- The usefulness of using the array as originally intended should be reviewed. The problems of minimum and maximum distances to defects should be determined, as well as the problems of matching a flat array to a rough, curved surface.

- The usefulness of having a steerable transmit beam should be reviewed. In might be better to have an entire array of receiving elements and a separate transmit element.

- If multiple transmitters are still felt to be useful, the ability to fire individual transmit elements should be included. This would greatly aid troubleshooting the system.